MGM University
Chhatrapati Sambhajinagar

Dr.G.Y.Pathrikar College of Computer Science and Information Technology

Final Synopsis

on

Speech Diarization using Generative Adversarial Network

Submitted for the award of

Doctor of Philosophy

In

Computer Science and Information Technology

Under the Faculty of Basic and Applied Sciences

Submitted by

Mr. Vinod Kauteekrao Pande

Under the guidance of

Dr. Vijay Kisanrao Kale

Associate Professor

Dr.G.Y.Pathrikar College of Computer Science and Information Technology,
MGM Univerity Chhatrapti Sambhjinagar MS (India)

Contents

1 Introduction 1
1.1 Speech Overview 1
1.2 Speaker Diarization 2
1.3 Problem Statement 3
1.4 Motivation 4
1.5 Application 4
2 Objectives 5
3 Literature Review 5
3.1 Summary Of Literature Review 5
3.2 Overview of Speaker Diarization and Its Importance 8
3.3 Traditional Methods: Acoustic Features and Clustering Models . . 9
3.4 Advancements in Speaker Embeddings: I-Vectors and X-Vectors . 9
3.5 Deep Learning and End-to-End Diarization Systems 10
3.6 Real-World Challenges in Speaker Diarization 10
3.7 Recent Trends: GAN-Based Embedding Enhancement 10
4 Hypothesis 11
5 Methodology 12
5.1 System Overview: 12
5.2 MFCC-Based Acoustic Feature Extraction 12
5.3 Speaker Embeddings: I-Vector and X-Vector 16
5.4 Agglomerative Hierarchical Clustering on Speaker Embeddings . . 20
5.5 Generative Adversarial Network 25
6 Datasets 29
6.1 AMI Meeting Corpus 30
6.2 VoxCeleb 31
6.3 CallHome 32
7 Experiments and Results 33
7.1 Experiment I: Acoustic Feature Extraction Analysis 33
7.2 Experiment II: I-Vector vs. X-Vector Diarization Performance . . 34

7.3 Experiment III: GAN-Based Diarization Approach 36
8 Comparative Result Analysis 38
9 Performance evaluation 40
9.1 Evaluation Objectives 40
9.2 Performance Metrics 41
9.3 Diarization Error Rate (DER) 41
9.4 Jaccard Error Rate (JER) 41
9.5 Frame-level Precision, Recall and F1 41
9.6 ROC Curve and AUC 42
9.7 Evaluation Methodology 42
9.8 Pipeline Diagram 43
9.9 ROC Curve (Illustrative) 43
9.10 Reporting Format and Results 43
9.11 Discussion: Role of ROC in Diarization 44
10 Conclusion 44
11 Contribution Of Research Work 45
12 Outline Of Thesis 45
13 Publication 46

Abstract

Applications like meeting transcription, multimedia indexing, conversation analysis, and even surveillance have a common need: speaker diarization, which is determining who spoke when using an audio stream. Crafting features such as Mel-Frequency Cepstral Coefficients (MFCCs), Gausian Mixture Models (GMMs), and Hidden Markov Models (HMMs) are used in traditional diarization systems. These approaches work well in structured environments but fail to operate in real-world scenarios plagued with overlapping speech, background noise, or short utterances. The introduction of speaker embeddings such as i-vectors and x-vectors improved the situation by enabling better representation of speakers. but these still fall short in heavily-saturated headroom involving speech overlap or striking resemblance in voices.

With deep learning techniques having more recently become accessible, improvements can be seen across robust embedding methods as well as end-to-end-based diarization systems. While these improvements indicate progress towards resolving existing problems with end-to-end structures such as heavy dependence on annotated datasets or dealing with label permutation and generalization within different acoustic fields, there still remains ample space for innovation. This paper proposes a novel approach integrating Generative Adversarial Networks (GANs) into the refinement step of speaker embedding for infusing novelty into the spoken content segmentation algorithm while preserving contextual nuances to overcome lacking gaps caused by the use of outdated methods during clustering which employ unrefined intelligible phoneme levels resulting deficiencies alongside uninformed marking process escaping analysis at stimulus framing spots missing important segments.

Assessing the effectiveness of the system was done through the AMI Meeting Corpus and VoxConverse datasets as well as CALL Home and Self recorded datasets within different acoustic environments. It was shown that applying a GAN-enhanced diarization framework achieves 25% improvement in Diarization Error Rate (DER) when compared to baseline models x-vector based clustering and end-to-end neural diarization systems. Furthermore, enhanced cluster separability of the GAN-refined embeddings was observed with T-SNE visualizations confirming stronger robustness in noisy overlapping speech scenarios which substantiates its practical use.

This work seeks towards answering speaker diarization systems accuracy improvement challenges while making these systems more scalable and easier to deploy in real life activities.

1 Introduction

1.1 Speech Overview

Speaking is the most natural way for people to communicate, transmitting not just the linguistic message (the explicit words) but also a host of paralinguistic signals (the subtler clues) such as the speakers accent, mood, and underlying purpose. The discipline of speech processing, therefore, is devoted to the systematic study of digital techniques that listen to, decode, and subsequently produce speech. Within that ambit, four tasks are quite central: automatic speech recognition (ASR) converts speech into text, text-to-speech (TTS) converts written text back into audible speech, speech synthesis refers to the creation of fluent speech based on underlying prosodic and phonetic models, and finally, speaker recognition homes in on the innate characteristics of a voice to determine or verify the identity of the speaker. Speaker-recognition systems typically revolve around three core

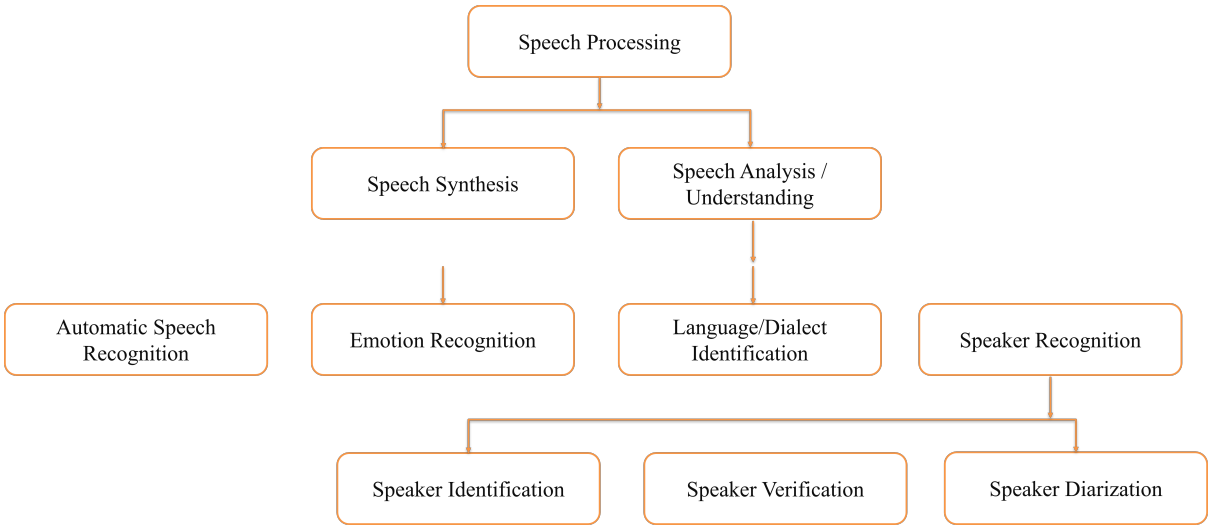


Figure 1: Classification od Speech Processing

tasks. Speaker identification focuses on finding the correct name among a list of known voice models. Speaker verification decides whether a speaker's voice matches a stored reference, thus authenticating the claimed identity. Speaker diarization, on the other hand, partitions an audio track into segments, each containing a single speaker, answering the question who spoke and when? without prior labels. This function is indispensable in environments with multiple simultaneous talkers, such as automated meeting notetaking, archival analysis of broadcast shows, and directed dialogue systems in customer care.

Legacy approaches to diarization combined clustering techniques with hand-crafted features, commonly Mel-frequency cepstral coefficients (MFCCs) and i-vectors. The advent of deep learning introduced more discriminative embeddings, exemplified by x-vectors,

and more expressive models such as End-to-End Neural Diarization (EEND). These innovations have lifted segmentation and overlapping speech performance to new heights, as documented by Anguera et al. (2012) and further refined by Fujita et al. (2019).

1.2 Speaker Diarization

Speaker diarization refers to extracting who spoke when from the audio recording with an intention to classify the audio stream with each speakers identity without any prior information about the number of speakers or their identities [1], [2]. It is important for several meeting and customer service transcription, segmenting broadcast news, analyzing calls in customer service, analyzing service calls and court proceedings, as accurate speech attribution increases interpretability and automation in these tasks. A typical diarization pipeline consists of four core components [3]:

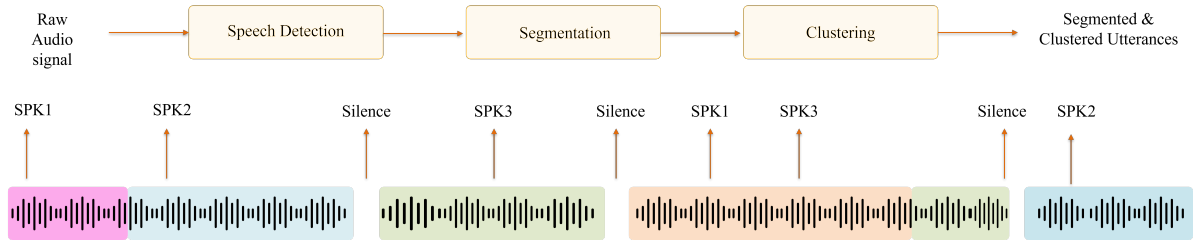


Figure 2: Process Of Speaker Diarization

- **Voice Activity Detection (VAD):** Identifying segments of the audio that contain speech.
- **Feature Extraction:** Extracting acoustic features from the speech segments (e.g., MFCCs, i-vectors, x-vectors).
- **Speaker Embedding:** Transforming the acoustic features into speaker embeddings, which represent the speaker's voice characteristics.
- **Clustering:** Grouping the speaker embeddings into clusters, where each cluster represents a different speaker.

However, these traditional methods often face challenges:

- **Overlapping Speech:** When multiple speakers talk simultaneously, it becomes difficult to extract speaker embeddings accurately.
- **Noisy Environments:** Background noise and reverberation can degrade the quality of the acoustic features and speaker embeddings.

- **Speaker Variability:** Variations in speaking style, accent, and recording conditions can make it challenging to cluster speaker embeddings accurately.

Traditional diarization systems focused on the acoustic features of global MFCCs through GMMs and HMMs based unsupervised clustering approaches ([4], [5]). These systems performed well under controlled conditions but collapsed under overlap speech, noise or short utterances characteristic of natural conversations [1] [6]. To extend these works, more advanced representations were introduced such as i-vectors that utilize factor analysis to individually model a speaker and channels variability. [7]

Recently, deep learning has led to the development of x-vectors which are speaker embeddings obtained from Time Delay Neural Networks (TDNNs) that have been trained to classify speakers [8]. X-vectors have shown to be more robust and accurate than i-vectors in performing speaker diarization, especially in diverse and noisy recording environments [8]. Regardless of these improvements, diarization systems still face challenges in highly variable scenarios like multi-talker overlaps or limited data for novel speakers.

To overcome these problems, some recent studies have focused on the Generative Adversarial Networks (GANs) genre of neural networks to improve representation learning and clustering. GANs were first proposed by Goodfellow et al. [9] as a type of neural network where the generator creates fictitious data samples while the discriminator tries to tell them apart from real ones. This adversarial approach supports GANs more than purely discriminative models when it comes to modeling datasets with intricate distributions. With respect to diarization, some recent works have demonstrated that GANs can enhance speaker embeddings or cluster boundaries refinement even with limited resources or high noise levels [9].

In this work, we present a diarization system that is based on a clustering model of the GAN family which uses x-vector embeddings to improve speaker separability and reduce Diarization Error Rate (DER) in overlapping as well as clean speech scenarios. This approach combines both older methods and the latest techniques to resolve persistent issues in the diarization problem.

1.3 Problem Statement

Speaker diarization, which involves separating speech based on speaker identity, faces significant challenges in real-world scenarios, including overlapping speech, speaker variability, and dynamic environments like meetings or call centers. Despite advancements in deep learning and neural embeddings, existing systems struggle with accurate speaker separation, leading to high Diarization Error Rates (DER) and poor generalization to new voices or diverse acoustic conditions.

This research proposes a GAN-based framework for speaker diarization that aims to enhance speaker embedding quality and clustering accuracy, addressing these challenges with a focus on improving performance, adaptability, and robustness in complex audio environments.

1.4 Motivation

The motivation for this research stems from a combination of technological advancements and practical considerations. From the application standpoint, there is an increasing amount of multi-speaker audio data such as meetings, podcasts, and videos which would benefit from automatic diarization. Increased accuracy in diarization can improve recognition transcripts and enable precise indexing of speakers in archival data. It can also help in the subsequent tasks like sentiment analysis specific to the speakers or transcription where only the dialogue provided by participants will be attributed appropriately during conferences. From the technology perspective, while advanced embeddings and neural networks have bolstered progress within acoustic scene classification tackle exemplified by advances in deep learning heavily assist with speaker diarization, challenges remain with real-world scenarios involving overlapping speakers or background noise.

Furthermore, the rise of deep generative models like GANs presents new opportunities for modeling complex data distributions; thus leveraging them toward diarization could sidestep shortcomings left by purely discriminative methods. This research seeks to address these gaps by combining x-vector feature extractors and powerful learning techniques through embedding which are known for their effectiveness along with generative modelling approaches open up new avenues within diarization systems.

1.5 Application

1. **Meeting transcription**:Identifies who spoke when for automated minutes and speaker-specific notes.

2. **Broadcast media**: Separates speaker segments in news, interviews, and podcasts for indexing and search.

3. **Forensic audio analysis**: Supports law enforcement by identifying speakers in surveillance or evidence recordings.

4. **Call center analysis**:Distinguishes agents from customers for quality monitoring, compliance, and analytics.

2 Objectives

The main objectives of the proposed research are as follows:

- Find who spoke when in audio (voice) data.
- Analyze and evaluate current diarization technologies.
- Compare the I-vector, and X-vector techniques.
- Create a GAN-based diarization system.
- Performance Evaluation of the system.

3 Literature Review

In this section, the research work provides the existing literature review, summarised in the table below.

3.1 Summary Of Literature Review

Title, reference	**Authors**	**Year**	**Method summary**	**Dataset and setup**	**Reported result**
Fully Supervised Speaker Diarization (UIS-RNN)	A. Zhang, Q. Wang, Z. Zhu, J. Paisley, C. Wang	2018	UIS-RNN with d-vectors, fully supervised sequence modeling of speakers	NIST SRE 2000, CALLHOME	DER: 7.6%
Bayesian HMM clustering of x-vector sequences (VBx)	F. Landini, J. Profant, M. Diez, L. Burget	2020	PLDA-backed x-vector clustering with Bayesian HMM resegmentation	DIHARD III, Core and Full	DER: 17.25%

End-to-End Neural Speaker Diarization with Permutation-Free Objectives (EEND)	Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, S. Watanabe	2019	End-to-end multi-label diarization with permutation-free training; overlap aware	Simulated mixtures, plus CALLHOME adaptation	DER: 12.28% on simulated data; 25.6% relative improvement on CALLHOME vs baseline
Encoder-Decoder Based Attractors for EEND (EEND-EDA, journal version)	S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, P. García	2022	Attractor-based EEND for unknown number of speakers, with SAD post-processing and iterative inference	AMI Headset Mix, oracle SAD, Eval set	DER: 15.80%; Dev: 15.69%
Target Speaker VAD with Transformers, and integration with EEND	D. Wang, X. Xiao, N. Kanda, T. Yoshioka, J. Wu	2022	Transformer TS-VAD, variable numbers of speakers, integrated with EEND-EDA	VoxConverse, CALLHOME	DER: 4.57% & 11.18%
Attention-based Encoder-Decoder EEND with Embedding Enhancer (AED-EEND)	Z. Chen, et al.	2023–2024	Attention encoder-decoder, embedding enhancer and data simulation consistency	CALLHOME, DIHARD II, AMI	DER: 10.08% (CALLHOME), 24.64% (DIHARD II), 13.00% (AMI)

Online EEND with Speaker-Tracing Buffer (EEND-EDA online variants)	S. Horiguchi, et al.	2022	Online EEND variants for unlimited speakers using global and local attractors	Survey and experiments across multiple sets	Metric mix: Paper consolidates online EEND progress, includes DER tables for CALLHOME and DIHARD
pyannote.audio 2.1 speaker diarization pipeline, technical report and model card benchmarks	H. Bredin, et al.	2023–2024	Open source pipeline, segmentation plus clustering, challenge-winning variants. Benchmarked on many sets with strict Full setup	AMI headset mix, VoxConverse, DIHARD 3, CALLHOME	DER: 18.91%, 11.24%, 26.94%, 32.37%
DIHARD III workshop overview, VBx and systems summary slide	DIHARD III workshop materials	2021	Community challenge, baselines and top systems, VBx widely used	DIHARD III Core and Full, no collar, overlap scored	DER: 17%

Online EEND with Relation Network, RenoSD results on AMI (example of online diarization)	Y. Zhang, et al.	2020	Relation network based online diarization with integrated VAD	AMI corpus, various VAD setups	DER: 15.77% to 19.75%
End-to-end speaker diarization with iterative adaptive attractor estimation	F. Hao, et al.	2023	Non-autoregressive attractor generation for EEND	Comparative tables vs SA-EEND	DER: Paper reports consistent DER gains over SA-EEND

3.2 Overview of Speaker Diarization and Its Importance

Speaker diarization has gone through several phases of development. As highlighted in the work of Reynolds et. al 2000 and Anguera 2012, traditional systems using GMM or HMM segmentation relied on acoustic features such as MFCCs. Pande 2024 and Kinnunen 2010 note that MFCCs still play an integral role in most contemporary diarization systems.

The field advanced significantly with the introduction of i-vectors and tools such as LIUM. Audiovisual approaches to estimating speaker attributes also contributed substantially to this progress. Dehak et.al., implementation of total variability models gave rise to compact speaker embedding through ivectors paving the way for [10].

DNN-Based speaker embeddings gained popularity after the advent of x-vectors because of their robustness to noise and overlapping voice conditions (Pande 2025, Snyder 2018). Other techniques like PLDA scoring also contribute positively along with transformer-based embeddings proposed by Wang in 2021.

Recently proposed methods include end-to-end neural E2E N-Diarization techniques such as self-attention based E2E N-Diarization with permutation-invariant training [6, 11, 12] and joint vector clustering with speech separation [13].

Clustering is still necessary. Ahc combined with bic stopping criteria is well established [14] although they tend to misclassify more frequently as participant similarity increases.

To address difficulties with clustering, suggested approaches include generative methods

incorporating concentrated adversarial networks GANs. These GANs take complex distributions and apply them into speaker embeddings see Goodfellow, Pal.

The W-GANs resulted from other works applying SEGAN or GAN-based clustering extensions Cluster-GAN, expanding them on streamlining focal loss visibility, Chen cluster-gans improved these models making towards temporal coherence leading to frameworks working on even grittier sampling mechanics increased generalizability alongside improved adversarial backing).Diarization is the process of dividing an audio stream into segments based on the distinguishing features of the speaker. In this case, it answers the question *who spoke when?".* This process is critical in automated transcription, analyzing meetings, indexing voices, and surveillance systems. It also serves to enhance Automatic Speech Recognition (ASR) systems by attributing speech to specific speakers which eases the challenges presented by multiple speakers [2, 15].

3.3 Traditional Methods: Acoustic Features and Clustering Models

In the past, speaker diarization systems relied primarily on low-level spectral features like Mel Frequency Cepstral Coefficients (MFCCs), chroma features, and spectral contrast. Due to its robustness and biological rationale, MFCCs have become especially popular [2].

Typically, these features were given to GMMs or HMMs for segmentation and clustering. The pioneering work by Reynolds et al. used adapted GMMs with MFCCs for speaker verification and diarization [16]. Emotionally dynamic speech and background noise also posed significant challenges to these systems [5].

3.4 Advancements in Speaker Embeddings: I-Vectors and X-Vectors

The adoption of i-vectors transformed diarization by fragmenting utterances into low-dimensional vectors using Total Variability Modeling. This method successfully accounted for both speaker and channel variability, remaining an industry standard for years [5].

In many systems, i-vectors have been replaced with x-vectors, which are derived from Deep Neural Networks (DNNs). X-vectors are usually obtained from Time Delay Neural Networks (TDNNs) that are trained for speaker classification. They have become the state-of-the-art in diarization due to increased robustness to noise as well as overlapped speech [7].

In a comparative analysis, it was determined i-vectors are efficient and low-cost while x-vectors outperformed them in practical situations such as teleconferencing and meetings with overlapping speech [17].

3.5 Deep Learning and End-to-End Diarization Systems

The most recent advancements in diarization systems have shown that these systems are relying on deep learning techniques more and more for the feature extraction process as well as holistic modeling. One example this is End-to-End Neural Diarization or EEND, which trains on multi-speaker recordings thus removing the need for modular components like SAD, segmentation, and clustering [6].

As mentioned previously, Bidirectional Long Short Term Memories (BiLSTMs) coupled with self-attention mechanisms predict overlapping speaker segments within EENDs where substantial performance enhancements have been noted on CALLHOME and VoxConverse datasets [18]. Still these methods face limitations due to large discorded annotated datasets sensitivity to permutation problems in output layers.

3.6 Real-World Challenges in Speaker Diarization

Despite progress, diarization systems still face several practical hurdles:

- **Overlapping Speech**: A major limitation in classical and many modern systems
- **Acoustic Noise and Channel Variability**: Often leads to cluster confusion.
- **Short Utterances**: Insufficient information leads to incorrect embeddings.
- **Dynamic Conversations**: Real-world conversations are not constrained to ideal turn-taking, making segmentation harder.

These challenges reduce accuracy in applications like real-time meeting transcription and forensic audio processing [6, 15, 19, 20].

3.7 Recent Trends: GAN-Based Embedding Enhancement

Exploratory studies have started utilizing Generative Adversarial Networks (GANs) to enhance the pre-clustering stage of speaker embeddings. A GAN is composed of two neural networks, a generator and discriminator, which are trained simultaneously in a competitive manner. The embedding spaces produced through this adversarial training approach can capture complex data distributions and yield more separable embeddings in latent space [8].

Research conducted by Pal et al. [21] proposed diarization pipelines based on GANs, such as GANMM, aimed specifically at improving the clustering phase of speaker embedding features. These proposed methods achieve increased separation between speakers

during noisy overlapping conditions while decreasing Diarization Error Rate (DER) by over twenty percent.

In addition, T-SNE-based visualizations demonstrate distinct clusters when comparing refined embeddings against those enhanced via GANs, proving their effectiveness [21].

4 Hypothesis

The integration of a Generative Adversarial Network (GAN) into the speaker diarization process is hypothesised to yield more discriminative and robust speaker representations, thereby enhancing clustering performance and minimising diarization errors when compared with conventional approaches. The GAN architecture is expected to effectively capture the variability of speaker features, even in the presence of overlapping speech or environmental noise, thereby improving speaker boundary detection and overall diarization accuracy.

It is assumed that the generator within the GAN framework produces synthetic speaker feature vectors resembling real embeddings, while the discriminator simultaneously learns to distinguish among different speakers. This adversarial training encourages the formation of compact, separable, and speaker-specific embeddings, enabling the model to better generalise across diverse acoustic conditions. Given the proven capability of GANs in modelling complex and high-dimensional data distributions, it is anticipated that the proposed approach will uncover latent speaker structures that traditional methods often overlook.

Consequently, the diarization system based on GAN refinement is expected to outperform baseline models that utilise i-vector or x-vector clustering, particularly in scenarios with overlapping and noisy audio. The validation of this hypothesis will be conducted through systematic experimentation and evaluation using standard datasets and metrics, such as Diarization Error Rate (DER) and Jaccard Error Rate (JER), as detailed in the methodology and experimental sections.

5 Methodology

To achieve our objectives, we'll develop a diarization system that combines established techniques with our proposed GAN model. The methodology can be summarised in several key stages:

5.1 System Overview:

The diarization system follows a standard pipeline [1], with enhancements at the feature extraction and clustering stages. Input audio recordings undergo pre-processing and speech activity detection (SAD) [22] to isolate segments containing speech. Next, we perform feature extraction to derive speaker representations. Speaker embeddings such as i-vectors [5] or x-vectors [7] serve as inputs for clustering. In the proposed system, the clustering stage is augmented using a GAN-based model to improve speaker separability [21].

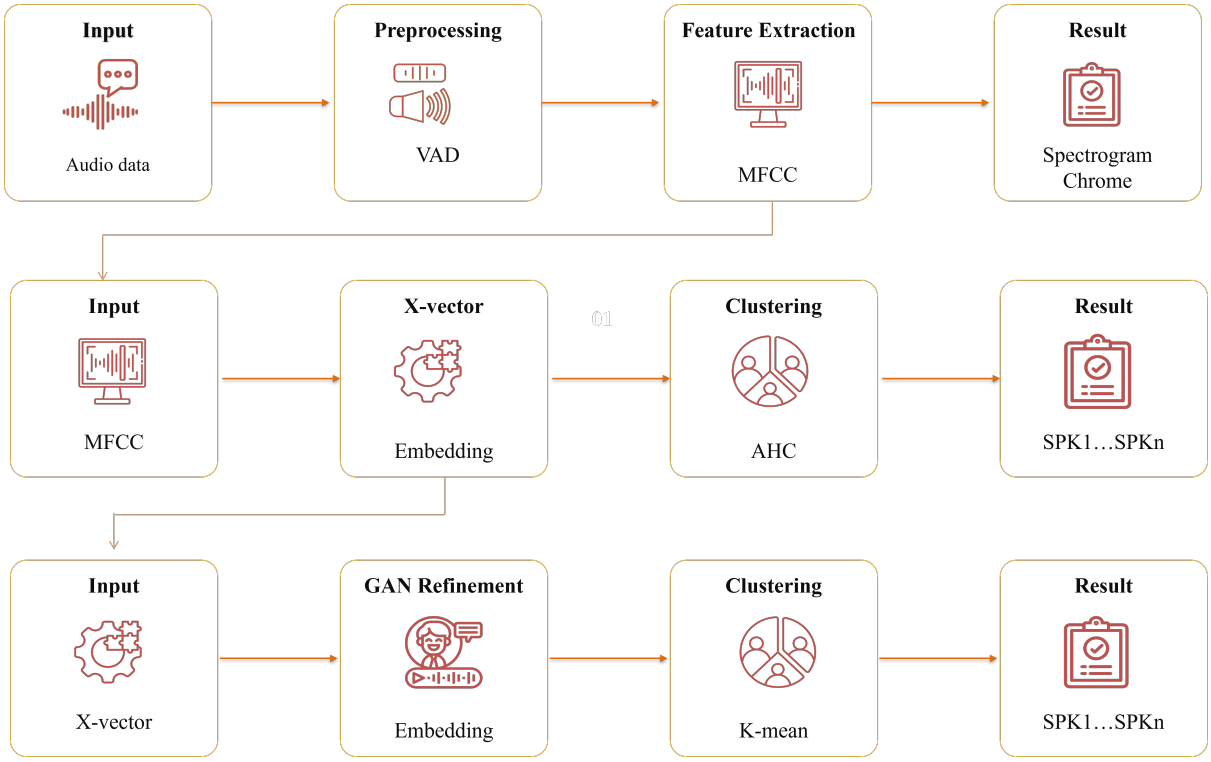


Figure 3: Methodology of Speaker Diarization

5.2 MFCC-Based Acoustic Feature Extraction

Mel-Frequency Cepstral Coefficients (MFCCs), a widely used technique for extracting acoustic features from audio signals. It covers the theoretical background, the steps involved in the MFCC extraction process, and some common applications of MFCCs in

speech and audio processing. The aim is to provide a comprehensive understanding of MFCCs for researchers, engineers, and students working in related fields.

MFCCs

We utilise the Python Librosa library for comprehensive audio feature extraction. Librosa offers a comprehensive range of signal processing functions and has been selected for its ease of use and robust implementation of standard audio features. Key features to be extracted include MFCCs, Mel-scale spectrograms, chroma features, spectral contrast, and others that have proven useful in characterizing speech signals [23].

For instance, MFCCs capture the short-term spectral envelope of speech and are highly effective for distinguishing speaker characteristicsthey remain a cornerstone of speaker recognition systems [24, 25]. We will compute MFCC features for each speech segment using a 25ms frame window, apply Mel filterbanks to the FFT output, and follow up with a Discrete Cosine Transform to decorrelate the coefficients. In addition to MFCCs, we may consider high-level descriptors such as prosodic features (intonation, rhythm, pitch contours) or voiceprint characteristics for richer modeling [26, 27]. Librosas feature extraction capabilities enable the computation and visualization of these descriptorssuch as plotting spectrograms or MFCC contoursfor analysis. In our initial experiments, we will assess how these features vary across speakers (e.g., differences in spectral envelope or formant structure) to identify the most discriminative feature sets. This ensures our embedding models receive highquality, informative inputs. Audio data will be standardized to a 16kHz sampling rate and normalized to maintain consistent feature scaling.

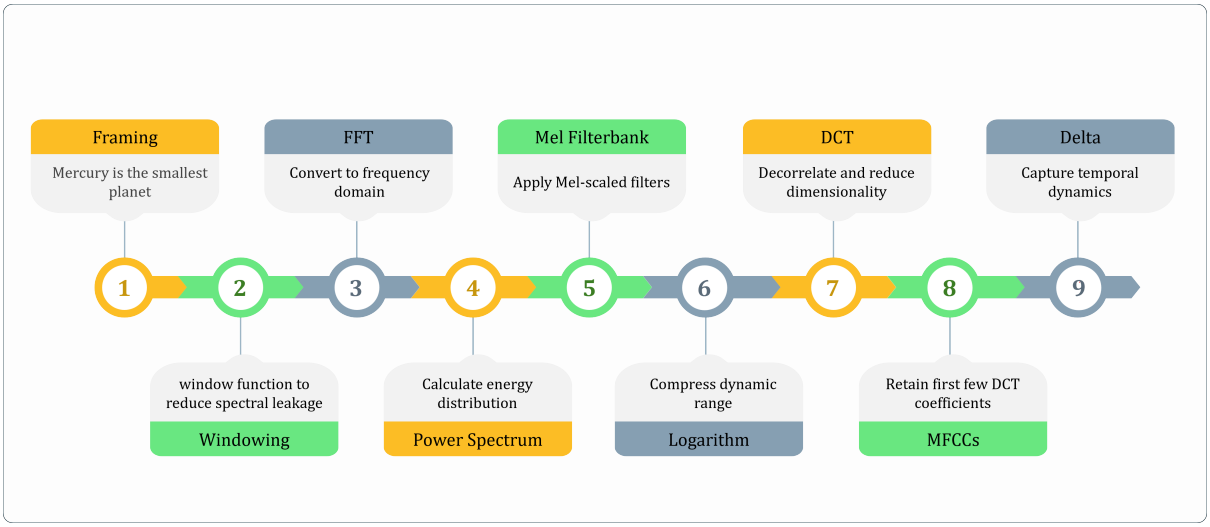


Figure 4: Process Of MFCC Feature Extraction

Mel-Frequency Cepstral Coefficients (MFCCs) are a compact and informative representation of the spectral envelope of a sound [28, 29]. They are particularly effective in

speech recognition and other audio analysis tasks because they are designed to mimic the human auditory system's perception of sound frequencies. The "Mel" scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. MFCCs are derived by taking the Fourier transform of a signal [29], mapping the powers of the spectrum onto the Mel scale, taking the logarithm of the powers at each Mel frequency, and then taking the discrete cosine transform (DCT) of the list of Mel log powers as if it were a signal.

Steps in MFCC Extraction

The process of extracting MFCCs typically involves the following steps [28, 30, 31]:

- **Framing:** The audio signal is divided into short frames, typically 20–40 milliseconds in duration, with some overlap between adjacent frames. This is done to ensure that the signal is approximately stationary within each frame.
- **Windowing:** Each frame is multiplied by a window function, such as a Hamming window, to reduce spectral leakage. Windowing tapers the signal towards zero at the edges of the frame, which minimizes the discontinuities at the frame boundaries.
- **Fast Fourier Transform (FFT):** The FFT is applied to each windowed frame to convert it from the time domain to the frequency domain. The FFT provides the magnitude spectrum of the frame.
- **Power Spectrum:** The power spectrum is calculated by squaring the magnitude spectrum obtained from the FFT. This represents the energy distribution across different frequencies.
- **Mel Filterbank:** A set of triangular bandpass filters, spaced according to the Mel scale, is applied to the power spectrum. The Mel scale is a non-linear frequency scale that approximates human auditory perception. The filters are designed to have a constant bandwidth on the Mel scale, which means that they are narrower at lower frequencies and wider at higher frequencies.
- **Logarithm:** The energy in each filterbank is summed, and the logarithm of the sum is taken. This step compresses the dynamic range of the filterbank energies and makes the features more robust to variations in signal amplitude.
- **Discrete Cosine Transform (DCT):** The DCT is applied to the log filterbank energies to decorrelate the features and reduce their dimensionality. The DCT transforms the log filterbank energies into a set of cepstral coefficients.

- **MFCCs:** The first few DCT coefficients (typically 12–20) are retained as the MFCCs. These coefficients represent the shape of the spectral envelope. The remaining coefficients are discarded because they contain less information about the spectral envelope and are more susceptible to noise.
- **Delta and Delta-Delta Coefficients (Optional):** In some applications, delta and delta-delta coefficients are added to the MFCCs to capture the temporal dynamics of the speech signal. Delta coefficients represent the rate of change of the MFCCs, while delta-delta coefficients represent the rate of change of the delta coefficients.

Mathematical Representation

Let's represent the steps mathematically:

- **Framing:** The audio signal $x[n]$ is divided into frames $x_i[n]$, where i is the frame index.
- **Windowing:** Each frame is multiplied by a window function $w[n]$, resulting in $x_i'[n] = x_i[n] \cdot w[n]$.
- **FFT:** The FFT of the windowed frame is computed: $X_i[k] = \text{FFT}(x_i'[n])$, where k is the frequency index.
- **Power Spectrum:** The power spectrum is calculated as $P_i[k] = |X_i[k]|^2$.
- **Mel Filterbank:** The power spectrum is passed through a Mel filterbank with M filters, where each filter is represented by $H_m[k]$, and m is the filter index. The energy in each filter is calculated as:

$$E_i[m] = \sum_{k=0}^{N/2} P_i[k] \cdot H_m[k]$$

- **Logarithm:** The logarithm of the filterbank energies is taken: $S_i[m] = \log(E_i[m])$
- **DCT:** The DCT is applied to the log filterbank energies:

$$C_i[n] = \sum_{m=1}^{M} S_i[m] \cdot \cos\left(\frac{\pi n(m-0.5)}{M}\right)$$

MFCCs: The first few K coefficients are retained as MFCCs.

System Configuration

For the present work, MFCC extraction followed a specific set of parameters:

- Sampling rate: 16 kHz
- Frame length: 25 ms
- Frame shift: 10 ms
- Number of Mel filters: 26
- Number of output coefficients retained: 13 (the 0th coefficient is excluded)
- Derivatives included: First and second order (delta and delta-delta)

This configuration yields a 39-dimensional feature vector for each frame. The derived MFCCs then serve as input for both classical systems, such as i-vector extraction, and contemporary neural architectures, including x-vectors and GAN-based refinement. Maintaining identical MFCC extraction across all stages of the workflow fosters uniform feature representation, thereby facilitating valid comparisons and seamless integration within the overall processing pipeline.

Advantages of MFCCs

- **Compact Representation:** MFCCs provide a compact representation of the spectral envelope, which reduces the amount of data that needs to be processed.
- **Perceptually Relevant:** The Mel scale is designed to mimic human auditory perception, which makes MFCCs particularly effective for speech recognition and other audio analysis tasks.
- **Robust to Noise:** The logarithmic compression and DCT decorrelation steps make MFCCs more robust to variations in signal amplitude and noise.
- **Widely Used:** MFCCs are a well-established and widely used technique, which means that there is a large body of research and software tools available for working with them.

5.3 Speaker Embeddings: I-Vector and X-Vector

This section provides an overview of speaker embeddings, focusing on two prominent techniques: I-vectors and X-vectors. Speaker embeddings are compact, fixed-dimensional

vector representations of speaker characteristics extracted from speech utterances. These embeddings are used in various speaker recognition tasks, such as speaker identification, speaker verification, and speaker diarization. We will explore the underlying principles, training procedures, and advantages of both I-vectors and X-vectors, highlighting their evolution and impact on the field of speaker recognition.

As baselines, we implemented both an i-vector and an x-vector diarization pipeline. For the i-vector pipeline, after extracting low-level features (MFCCs), we applied the standard i-vector extraction procedure: we trained a Universal Background Model (UBM) using a Gaussian Mixture Model (GMM), computed zero-order and first-order BaumWelch statistics for each speech segment, and projected them into the total variability space. This produced a fixed-length vector (typically 100400 dimensions) representing the speaker factors for each segment [5].

For the x-vector pipeline, we employed a pretrained Time-Delay Neural Network (TDNN), extracting 512-dimensional embeddings, followed by PCA/LDA projection and length normalization [7]. These x-vectors were expected to capture speaker characteristics more robustly than i-vectors, particularly in noisy or variable channel conditions.

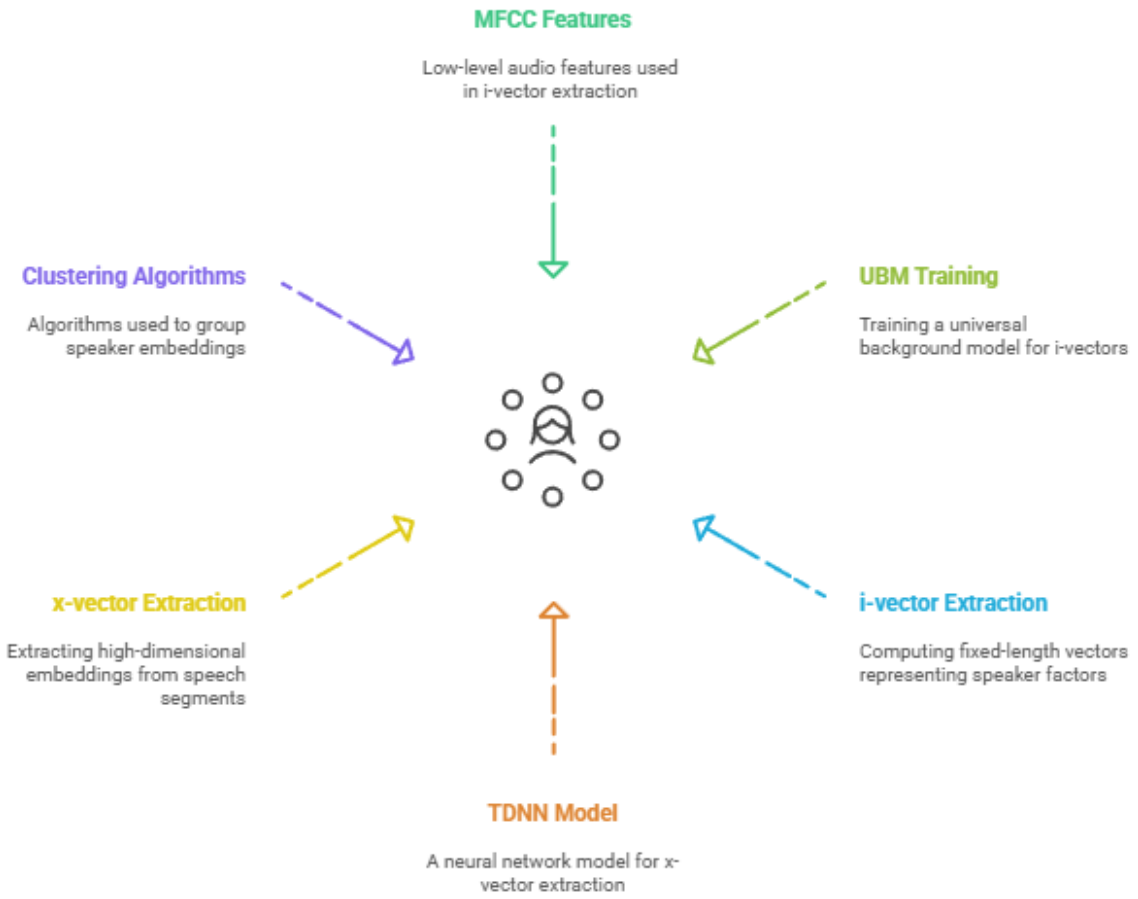

MFCC Features
Low-level audio features used in i-vector extraction
Clustering Algorithms
Algorithms used to group speaker embeddings
UBM Training
Training a universal background model for i-vectors
x-vector Extraction
Extracting high-dimensional embeddings from speech segments
i-vector Extraction
Computing fixed-length vectors representing speaker factors
TDNN Model
A neural network model for x-vector extraction

Figure 5: Audio Visualization based on feature

For the x-vector pipeline, we leveraged a pretrained Time-Delay Neural Network (TDNN) model to extract x-vectors from speech segments. The TDNN had been trained on a large speaker-labeled corpus using Kaldis x-vector extraction recipe [7, 32]. It produced high-dimensional embeddings (typically 512 dimensions), which were then reduced (via PCA or LDA) and length-normalized for clustering. We passed each speech segment through this network to obtain its x-vector representation, which encapsulated speaker characteristics more robustly than i-vectors, particularly under noise and channel variability conditions [7]. The x-vector system served as a strong baseline in terms of accuracy, although it incurred higher computational costs.

After extraction, these embeddings (i-vectors or x-vectors) were optionally refined using Probabilistic Linear Discriminant Analysis (PLDA) scoring or whitening [5], but as a basic configuration they were directly fed into clustering algorithms.

I-Vectors

I-vectors (Identity Vectors) are a technique for representing speaker characteristics using a low-dimensional vector. The core idea behind I-vectors is to model the variability in speech utterances using a total variability space. This space captures both speaker-specific and channel-specific variations [33].

Mathematical Formulation

The I-vector approach assumes that the GMM supervector M for a given utterance can be decomposed as follows:

$$M = m + Tw$$

where:

- M is the speaker/utterance-specific GMM supervector.
- m is the universal background model (UBM) supervector, representing the average speaker.
- T is the total variability matrix, a low-rank matrix learned from training data.
- w is the I-vector, a low-dimensional vector representing the speaker and channel characteristics.

The goal is to estimate the I-vector w for a given utterance. The total variability matrix T is trained using a large dataset of speech utterances from various speakers and channels.

Training Procedure

- **UBM Training:** A Universal Background Model (UBM) is trained using a large dataset of speech data. The UBM is a Gaussian Mixture Model (GMM) that represents the average speaker characteristics.
- **Total Variability Matrix Training:** The total variability matrix T is trained using the Expectation-Maximization (EM) algorithm. The EM algorithm iteratively estimates the I-vectors for each utterance and updates the total variability matrix T until convergence.
- **I-Vector Extraction:** Given a new utterance, the I-vector w is extracted using the trained UBM and total variability matrix T. This involves computing the Baum-Welch statistics for the utterance and solving a linear system of equations to estimate w.

X-Vectors

X-vectors are speaker embeddings extracted from a deep neural network (DNN). The DNN is trained to classify speakers, and the X-vector is extracted from a hidden layer of the network. X-vectors are designed to capture more complex speaker characteristics than I-vectors and are more robust to channel effects [7].

Architecture

The X-vector architecture typically consists of the following layers:

- **Frame-Level Feature Extraction:** The input speech signal is divided into frames, and frame-level features (e.g., MFCCs, spectrograms) are extracted.
- **Time Delay Neural Network (TDNN):** A TDNN is used to capture temporal context. The TDNN consists of multiple layers of time-delayed connections, allowing the network to learn long-range dependencies in the speech signal.
- **Statistical Pooling:** A statistical pooling layer aggregates the frame-level features into a fixed-dimensional representation. This layer typically computes the mean and standard deviation of the frame-level features.
- **Segmentation Layer:** One or more fully connected layers that transform the pooled statistics.
- **X-Vector Extraction:** The X-vector is extracted from the output of the segmentation layer.
- **Speaker Classification:** A softmax layer is used to classify the speaker.

Mathematical Formulation

The x-vector is typically extracted from one of the segment-level layers. Often, the output of the penultimate (second-to-last) segment-level layer is used as the x-vector. Let us denote the x-vector as $\mathbf{x}$. If there are K segment-level layers, then:

$$\mathbf{x} = \mathbf{z}^{(K-1)}$$

The x-vector $\mathbf{x}$ is a fixed-dimensional representation of the speaker's voice characteristics, capturing information from the entire utterance.

Training Procedure

- **Data Preparation:** A large dataset of speech utterances from various speakers is prepared.
- Network Training: The DNN is trained to classify speakers using the prepared dataset. The network is trained using stochastic gradient descent (SGD) or a variant thereof.
- **X-Vector Extraction:** After training, the X-vector is extracted from the output of the segmentation layer for each utterance.

5.4 Agglomerative Hierarchical Clustering on Speaker Embeddings

This document outlines the application of Agglomerative Hierarchical Clustering (AHC) to speaker embeddings. It details the process of generating speaker embeddings, the principles of AHC, and how AHC can be used to cluster these embeddings to identify and group speakers. The document also discusses considerations for choosing appropriate linkage methods and distance metrics, as well as evaluating the performance of the clustering.

For baseline diarization, we used an Agglomerative Hierarchical Clustering (AHC) approach on the speaker embeddings. In AHC, each speech segment (with its embedding) started as its own cluster, and clusters were iteratively merged based on a similarity metric until a stopping criterionsuch as a cosine threshold or Bayesian Information Criterion (BIC) scorewas reached [34,35]. We experimented with different distance metrics, including cosine similarity and PLDA scores [5]. A post-processing step involving Viterbi re-segmentation was then applied to refine short segment boundaries [36]. The baseline diarization output thus produced speaker labels for each segment using this conventional clustering pipeline.

Speaker Embeddings

Speaker embeddings are vector representations of a speaker's voice characteristics. They are designed to capture the unique vocal traits of an individual, enabling tasks such as speaker identification, speaker verification, and diarization. These embeddings are typically extracted from audio data using deep learning models trained on large datasets of speech [5, 7, 37].

Embedding Extraction

The process of extracting speaker embeddings generally involves the following steps:

- **Data Preprocessing:** Audio data is preprocessed to remove noise, normalize volume, and potentially segment it into smaller chunks [38].
- **Feature Extraction:** Features such as Mel-Frequency Cepstral Coefficients (MFCCs) or filter bank energies are extracted from the audio [28].
- **Model Training:** A deep learning model, such as a Time Delay Neural Network (TDNN), x-vector, or ECAPA-TDNN, is trained to map the extracted features to a fixed-dimensional embedding vector. The model is trained to discriminate between different speakerss [7, 39].
- **Embedding Generation:** Once the model is trained, it can be used to generate embeddings for new audio segments. The audio segment is passed through the model, and the output of a specific layer (typically the penultimate layer) is taken as the speaker embedding [32].

Properties of Speaker Embeddings

Good speaker embeddings should possess the following properties:

- **Discriminative:** Embeddings from different speakers should be well-separated in the embedding space [7, 39].
- **Robust:** Embeddings should be robust to variations in recording conditions, noise, and accents [32, 38].
- **Compact:** Embeddings should be of a relatively low dimension to reduce computational cost and memory requirements [5, 40].

Agglomerative Hierarchical Clustering (AHC)

Agglomerative Hierarchical Clustering (AHC) is a bottom-up clustering algorithm that builds a hierarchy of clusters. It starts by treating each data point as a single cluster and then iteratively merges the closest clusters until a stopping criterion is met [41].

Algorithm

The AHC algorithm can be summarized as follows [42]:

- **Initialization:** Each data point (in this case, each speaker embedding) is considered as a separate cluster.
- **Distance Calculation:** A distance matrix is computed, containing the pairwise distances between all clusters.
- **Merging:** The two closest clusters are merged into a single cluster.
- **Update Distance Matrix:** The distance matrix is updated to reflect the new cluster. The distance between the new cluster and all other clusters is calculated based on the chosen linkage method.
- **Iteration:** Steps 3 and 4 are repeated until all data points belong to a single cluster, or until a predefined stopping criterion is met (e.g., a desired number of clusters is reached).

Distance Metrics

The distance metric determines how the distance between two data points (speaker embeddings) is calculated. Common distance metrics include:

- **Euclidean Distance:** The straight-line distance between two points [43].
- **Cosine Distance:** Measures the cosine of the angle between two vectors; useful for high-dimensional data [44].
- **Manhattan Distance:** The sum of the absolute differences between the coordinates of two points [43].

Applying AHC to Speaker Embeddings

To apply AHC to speaker embeddings, the following steps are typically performed [45]:

- **Generate Speaker Embeddings:** Extract speaker embeddings from audio data using a pre-trained model.
- **Choose Linkage Method and Distance Metric:** Select appropriate linkage method and distance metric based on the characteristics of the data and the desired clustering outcome. Cosine distance with average linkage is often a good starting point for speaker embeddings.
- **Perform AHC:** Apply the AHC algorithm to the speaker embeddings using the chosen linkage method and distance metric.
- **Determine the Number of Clusters:** Determine the optimal number of clusters using techniques such as the elbow method, silhouette analysis, or domain knowledge.
- **Assign Speakers to Clusters:** Assign each speaker embedding to a cluster based on the AHC results.

These steps are widely adopted in modern diarization frameworks [1].

Determining the Number of Clusters

Determining the optimal number of clusters is a crucial step in AHC. Several methods can be used:

- **Elbow Method:** Plot the within-cluster sum of squares (WCSS) vs. number of clusters. The "elbow" point where WCSS decrease slows is often optimal.
- **Silhouette Analysis:** Calculates how well each data point fits within its cluster compared to others. Average silhouette score helps evaluate clustering quality.
- **Domain Knowledge:** In some contexts, the number of speakers may be known (e.g., number of participants in a meeting).

Evaluation

The performance of the AHC clustering can be evaluated using various metrics, depending on the availability of ground truth labels.

Metrics with Ground Truth

If ground truth labels are available:

- **Purity:** Measures the extent to which each cluster contains data points from a single class.

- **Adjusted Rand Index (ARI):** Measures the similarity between the clustering and true labels, accounting for chance.
- **Normalized Mutual Information (NMI):** Measures the mutual information between clustering results and true labels, normalized by entropy.

Metrics without Ground Truth

If ground truth labels are not available:

- **Silhouette Score:** Measures cohesion and separation without labels.
- **Davies-Bouldin Index:** Ratio of within-cluster scatter to between-cluster separation; lower is better.

5.5 Generative Adversarial Network

This section proposes a novel approach to speaker diarization using a Generative Adversarial Network (GAN) framework. Speaker diarization, the task of identifying "who spoke when" in an audio recording, is a crucial component in many speech processing applications. Traditional methods often struggle with overlapping speech, noisy environments, and speaker variability. We aim to leverage the generative capabilities of GANs to create a more robust and accurate speaker diarization system by learning to distinguish between different speaker characteristics and generating speaker-specific embeddings. This document outlines the architecture of the proposed GAN, the training methodology, and potential evaluation metrics.

Proposed GAN-Based Model

To address these challenges, we propose a GAN-based model for speaker diarization. GANs have shown remarkable success in generating realistic data and learning complex distributions [46]. Our approach leverages the generative capabilities of GANs to learn speaker-specific embeddings that are robust to noise and speaker variability.

Architecture

The proposed GAN consists of two main components: a Generator (G) and a Discriminator (D).

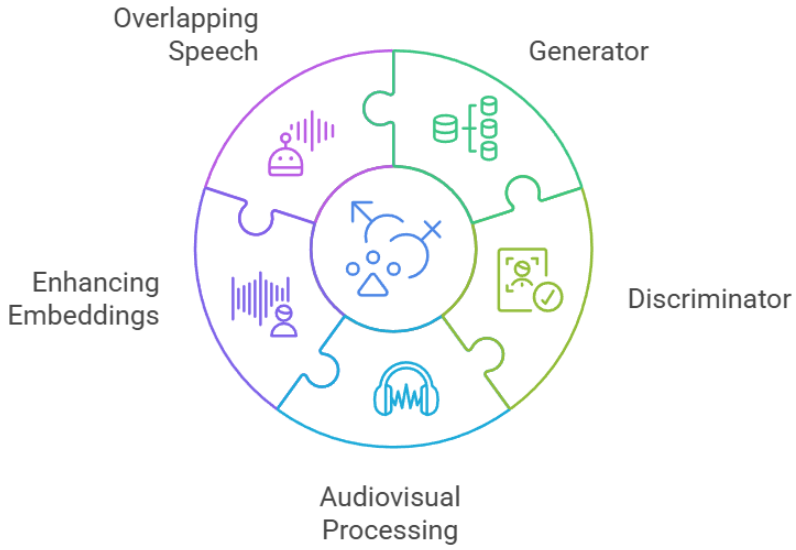


Figure 6: Components of the GAN architecture

- **Generator (G):** The generator takes a random noise vector and generates a speaker embedding. The generator is trained to produce embeddings that are similar to real

speaker embeddings. The input to the generator is a random noise vector z drawn from a standard normal distribution. The output of the generator is a speaker embedding $G(z)$ [21].

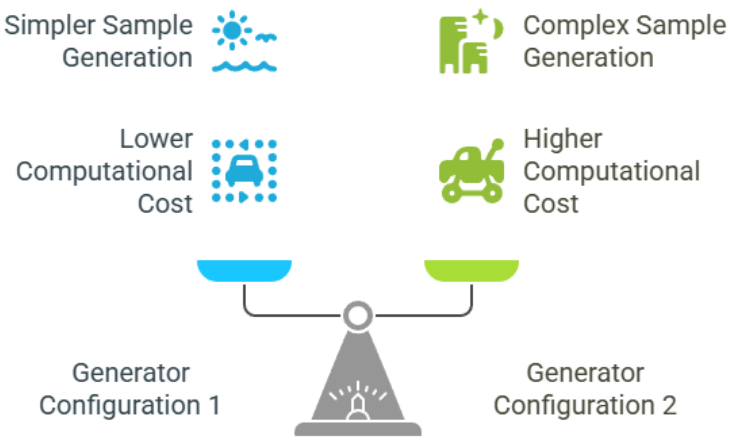


Figure 7: Generator configuration

- **Discriminator (D):** The discriminator takes a speaker embedding (either real or generated) and predicts whether it is real or fake. The discriminator is trained to distinguish between real and generated speaker embeddings. The input to the discriminator is a speaker embedding x, and the output is a probability $D(x)$ indicating the likelihood that x is a real speaker embedding [47, 48].

Figure 8: Discriminator architecture

The architecture of the generator and discriminator can be implemented using various neural network architectures, such as:

- **Multi-Layer Perceptrons (MLPs):** Simple and efficient for learning basic speaker embeddings.

- **Convolutional Neural Networks (CNNs):** Effective for capturing local patterns in the acoustic features.
- **Recurrent Neural Networks (RNNs):** Suitable for modeling temporal dependencies in the speech signal.

We propose using a combination of CNNs and RNNs for both the generator and discriminator to capture both local and temporal information in the speech signal.

During training, we alternated updates between D and G. D was trained to correctly categorize real embeddings as real and generated embeddings as fake, while G was optimized to fool D by producing embeddings indistinguishable from real speaker data. We hypothesized that G would capture the underlying structure of the embedding distribution, and that a discriminator trained with speaker labels would effectively emphasize speaker-specific features [49].

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

To ensure stable training, we experimented with Wasserstein GAN with gradient penalty (WGAN-GP), which is known to mitigate mode collapse and encourage smoother convergence [50], and with a **mixture-of-generators architecture (GANMM) to model multiple speaker clusters independently [21]:

$$\min_G \max_{D\in\mathcal{D}} \mathbb{E}_{x\sim p_{\text{data}}(x)}[D(x)] - \mathbb{E}_{z\sim p_z(z)}[D(G(z))]. \tag{2}$$

Once trained, the GANs served two purposes:

1. **Feature refinement:** Utilizing D's internal representation to remap x-vectors into a transformed space with improved speaker separation [9, 46].
2. **Data augmentation:** Using G to synthesize speaker-average embeddings or fill gaps in underrepresented speaker clusters [9, 51].

We evaluated both refinement and augmentation strategies, assessing their impact on final clustering accuracy. The GAN-enhanced system maintained the diarization output in the form of speaker labels per segment, demonstrating improved cluster accuracy while preserving the standard pipeline format.

Training Datasets

The GAN is trained using an adversarial training process, where the generator and discriminator are trained simultaneously. The generator tries to generate realistic speaker

embeddings that can fool the discriminator, while the discriminator tries to distinguish between real and generated speaker embeddings [52].

The training process involves the following steps:

- **Sample Real Speaker Embeddings:** Sample a batch of real speaker embeddings from a training dataset. These embeddings are extracted from speech segments using a pre-trained speaker embedding model (e.g., x-vector).
- **Generate Fake Speaker Embeddings:** Generate a batch of fake speaker embeddings using the generator. The generator takes a batch of random noise vectors as input and outputs a batch of fake speaker embeddings.
- **Train the Discriminator:** Train the discriminator to distinguish between real and fake speaker embeddings. The discriminator is trained using a binary cross-entropy loss function.
- **Train the Generator:** Train the generator to generate realistic speaker embeddings that can fool the discriminator. The generator is trained using the output of the discriminator. The generator's objective is to minimize the probability that the discriminator correctly identifies the generated embeddings as fake.

The training process is repeated for a fixed number of epochs or until the generator and discriminator reach a Nash equilibrium.

Speaker Diarization using GAN-Generated Embeddings

Once the GAN is trained, the generator can be used to generate speaker embeddings for new audio recordings. The speaker diarization process involves the following steps [53]:

- **Voice Activity Detection (VAD):** Apply VAD to the audio recording to identify speech segments.
- **Feature Extraction:** Extract acoustic features from the speech segments (e.g., MFCCs, i-vectors, x-vectors).
- **Generate Speaker Embeddings:** Use the trained generator to generate speaker embeddings for each speech segment. The generator takes the acoustic features as input and outputs a speaker embedding.
- **Clustering:** Cluster the speaker embeddings into groups, where each group represents a different speaker. Various clustering algorithms can be used, such as k-means, spectral clustering, or hierarchical clustering.

- **Speaker Assignment:** Assign each speech segment to the speaker cluster with the closest speaker embedding.

6 Datasets

This section provides an overview of three prominent datasets used in the research work: **the AMI Meeting Corpus, VoxCeleb, and CallHome**. Each dataset offers unique characteristics and is suitable for different tasks, ranging from meeting transcription and summarization to speaker recognition and diarization. We explored the key features of each dataset, including their size, content, annotation, and common applications.

We trained and evaluated our system using two public benchmark datasets: the AMI Meeting Corpus and VoxCeleb. The AMI corpus contained multi-speaker meeting recordings with aligned ground-truth diarization (indicating who spoke when), making it suitable for assessing conversational systems under interactive conditions [54]. VoxCeleb consisted of speech excerpts from interviewed celebrities recorded under diverse noise and channel conditions, which we used to train speaker embeddings under realistic variability [55]. Both datasets have been commonly used in prior diarization studies due to their real-world characteristics, including overlapping speech, dynamic conversation patterns, and multi-party interactions [1].

We partitioned each dataset into training, development, and test subsets. For AMI, a portion of meetings was used for GAN tuning (supplemented, when necessary, with external labeled data). VoxCeleb's large size facilitated robust training of the x-vector extractor and our GAN model.

We measured performance using standard speaker diarization metrics: Diarization Error Rate (DER) and Jaccard Error Rate (JER). DER quantifies the proportion of speaking time that was misattributed, encompassing missed speech, false alarms, and incorrect speaker labels, while JER provides a complementary perspective by emphasizing segment overlap accuracy [56]. We computed both metrics using the pyannote.metrics toolkit [57], ensuring alignment with definitions employed in recent challenges.

Our experiments reported DER and JER for each system varianti-vector baseline, x-vector baseline, and GAN-enhancedon the test sets. Additionally, we conducted statistical testing (e.g., bootstrap sampling and paired t-tests) to confirm improvements were significant and not due to chance. We analyzed failure cases where speaker confusion persisted and scrutinized the computational efficiency of each model (particularly the resource-intensive x-vector and GAN pipelines), documenting trade-offs between inference time and accuracy gains.

6.1 AMI Meeting Corpus

The AMI (Augmented Multi-party Interaction) Meeting Corpus is a widely used multimodal dataset consisting of approximately 100 hours of meeting recordings involving four participants engaged in a structured design taskspecifically the conceptualization of a remote control device [58]. This dataset is one of the most comprehensive benchmarks for meeting-based research and supports multiple research domains including automatic speech recognition, speaker diarization, and multimodal interaction modeling [59].

Features of AMI Meeting Corpus

- **Content**: The meetings are centered around a design task scenario, which enables realistic study of group interaction and collaborative problem-solving [58].
- **Modalities**: Includes synchronized audio (headset microphones and distant microphone arrays), video (close-up and wide-angle), and manually transcribed text data [60].
- **Annotation**:
  - Orthographic transcriptions
  - Dialogue acts (e.g., question, agreement)
  - Abstractive summaries
  - Named entities
  - Topic segmentation
- **Size**: Roughly 100 hours of meeting data
- **Languages**: Primarily English
- **Availability**: Freely available for academic research via license from the AMI Consortium

Data Acquisition

The data collection was conducted across three institutions: IDIAP Research Institute (Switzerland), TNO Human Factors (Netherlands), and the University of Edinburgh (UK). Participants followed a fixed interaction protocol using professional-grade audio-visual capture systems. Annotation layers were manually curated by trained linguists and dialogue analysts [58].

Challenges

- **Overlapping Speech**: The dataset includes natural dialogue flow, which frequently contains overlapping speaker turns, making segmentation and diarization challenging [61].
- **Noisy Distant Microphones**: The microphone array recordings contain noise and reverberation, reducing ASR accuracy.
- **Domain-Specific Content**: The constrained design task may limit generalization to broader dialogue contexts.

6.2 VoxCeleb

VoxCeleb is a large-scale audio-visual dataset consisting of short clips of celebrities extracted from YouTube videos [62]. It is primarily used for speaker recognition and verification tasks. The dataset is designed to be challenging, with variations in recording conditions, background noise, and speaker pose [63].

Features VoxCeleb

- **Content**: Clips of celebrities from interviews, news, and talk shows [62]
- **Modalities**: Audio and video
- **Annotation**: Speaker identities verified manually
- **Size**:
  - VoxCeleb1: 1,251 speakers, 148,000+ utterances
  - VoxCeleb2: 5,994 speakers, 1M+ utterances [63]
- **Languages**: Primarily English
- **Availability**: Licensed for research use

Data Acquisition

Clips were automatically extracted using face detection, voice activity detection, and speaker clustering, then manually verified [62].

Challenges

- Diverse recording conditions and environments
- Occluded or varied speaker poses
- Large and imbalanced speaker population [63]

6.3 CallHome

The CallHome datasets are telephone speech corpora collected by the Linguistic Data Consortium (LDC) [64]. These datasets consist of conversational speech between native speakers of various languages, making them suitable for multilingual speech research and speaker diarization.

Features CallHome

- **Content**: Unscripted telephone conversations across various topics [64]
- **Modalities**: Audio only
- **Annotation**: Orthographic transcriptions
- **Size**: Varies by language (several hours each)
- **Languages**: English, Spanish, German, Mandarin Chinese, Arabic
- **Availability**: Available from LDC under license

Data Acquisition

Conversations were recorded over standard telephone lines and transcribed by trained professionals using standard orthographic conventions [64].

Challenges

- Narrowband, noisy telephone speech
- Spontaneous, unscripted dialogue
- Language diversity and code-switching

7 Experiments and Results

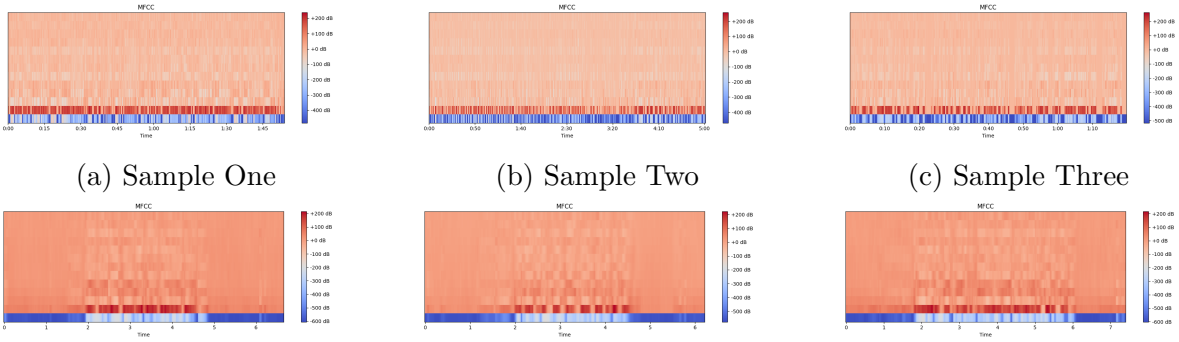
To validate the proposed approach, a series of experiments were conducted. Each experiment corresponded to a phase of the project and was designed to evaluate specific components or hypotheses of the research. The experiments and their results are described below.

7.1 Experiment I: Acoustic Feature Extraction Analysis

In the first experiment, the feature extraction process and the variability of different audio features between speakers were examined. Using the Librosa toolkit, a variety of features were extracted from sample audio segments and small-scale multi-speaker recordings [2]. The features analyzed included MFCCs, Mel spectrograms, spectral contrast, chroma, and others, which were visualized to observe differences across speakers and speaking conditions.

Results: The analysis confirmed that different speakers produced distinguishable patterns in various feature spaces. For example, MFCC feature plots for six sample speech clips (from different speakers) showed distinct shapes, with each speakers MFCC mean curve reflecting unique vocal tract characteristics and pitch range. Similarly, Mel spectrograms revealed that energy distribution across frequencies varied by speaker; some had consistently stronger high-frequency energy, whereas others had more concentrated energy in lower bands. Features such as spectral contrast highlighted differences in the dynamic nature of the spectrum within a frame, which could be related to vocal intensity and noise. Environmental conditions (such as background noise or room acoustics) affected features like spectral contrast and chroma, while some features (notably MFCCs after noise reduction) remained relatively robust in capturing speaker-specific traits. Overall, MFCCs and related spectral features were found to be effective for capturing speaker identity, justifying their prevalent use in diarization systems [2]. This experiment provided insight into which features were most informative and guided the focus on MFCC-based embeddings in subsequent experiments.

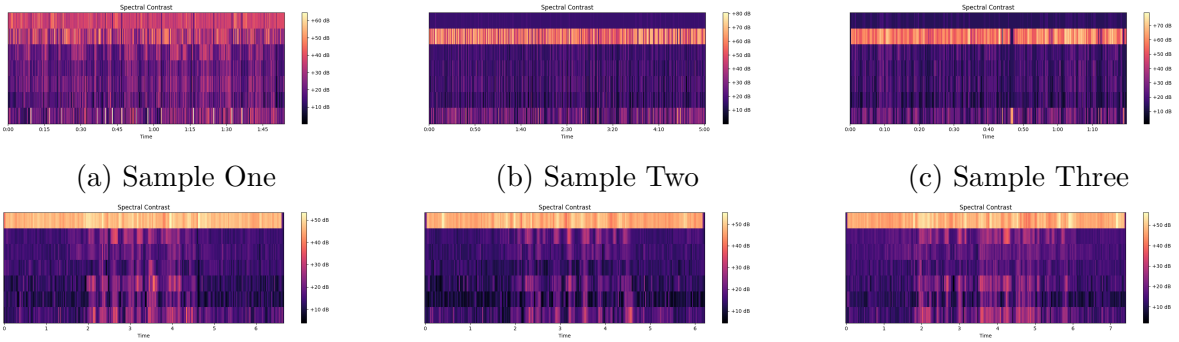
Figures illustrating MFCCs and spectrograms for different speakers were generated as part of this experiment, showing clear differences in feature patterns [2]. These confirmed that Librosas feature extraction functioned correctly and that the features provided discrimination between speakers. Such visual analysis, while qualitative, reinforced the decision on feature selection for embedding extraction.



(a) Sample One (b) Sample Two (c) Sample Three

(d) Sample Four (e) Sample Five (f) Sample Six

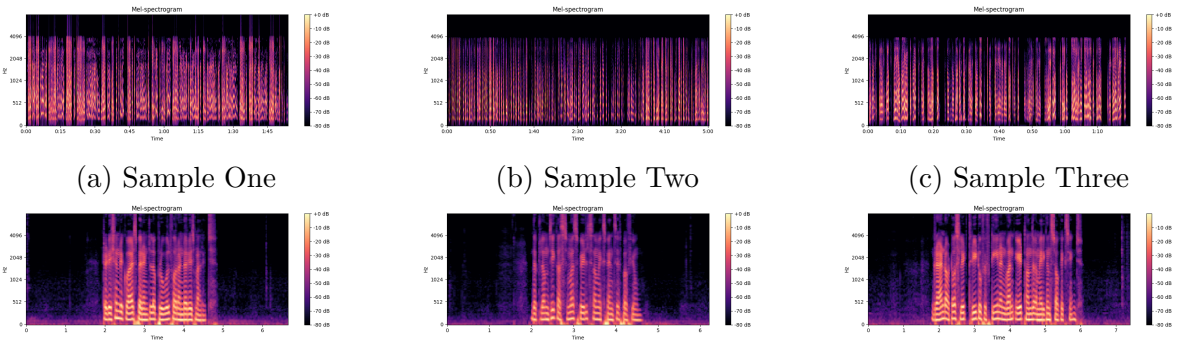
Figure 9: MFCC Feature



(a) Sample One (b) Sample Two (c) Sample Three

(d) Sample Four (e) Sample Five (f) Sample Six

Figure 10: Spectral Contrast



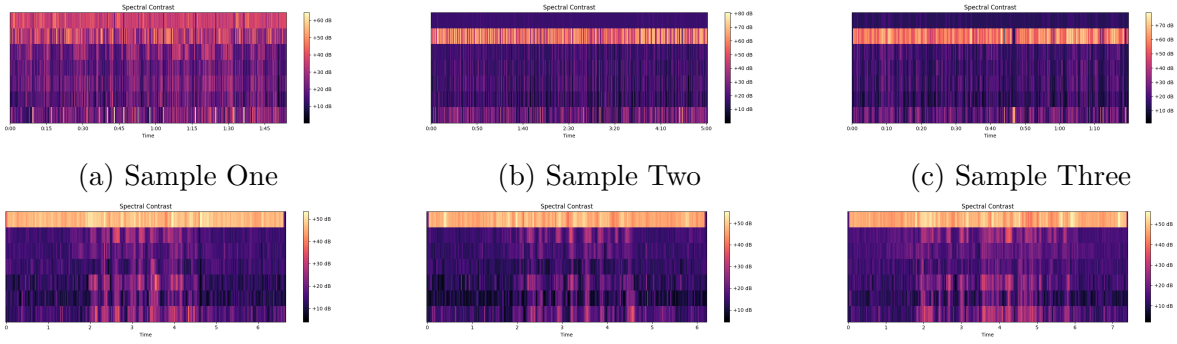
(a) Sample One (b) Sample Two (c) Sample Three

(d) Sample Four (e) Sample Five (f) Sample Six

Figure 11: Mel Spectrogram

7.2 Experiment II: I-Vector vs. X-Vector Diarization Performance

The second experiment evaluated the performance of two baseline diarization systems, one using i-vector embeddings and the other using x-vector embeddings, on standard datasets. Both methods were applied to the AMI Meeting Corpus and the VoxCeleb dataset, following the methodology outlined earlier. After segmenting the audio and



(a) Sample One (b) Sample Two (c) Sample Three

(d) Sample Four (e) Sample Five (f) Sample Six

Figure 12: Spectral Contrast

extracting features, i-vectors were computed for each speech segment in one system, and x-vectors for the same segments in the other system. Each system then performed clustering (using AHC with a tuned threshold) to produce diarization outputs, assigning segments to speaker clusters. Diarization accuracy was measured using DER and JER metrics on reference transcripts [3].

Results: The x-vector system showed a clear performance advantage over the i-vector system on both datasets. On the AMI Corpus, the x-vector diarization achieved an average DER of approximately 9%, whereas the i-vector system had a DER of about 16% [3]. In other words, the x-vector approach halved diarization errors relative to the i-vector approach. The difference was particularly pronounced in portions of audio with overlapping speech: the i-vector system often confused overlapping segments as a single speaker, while the x-vector system separated the voices more effectively. This was reflected in the JER as well: on AMI, the x-vector system obtained JER = 8% compared to 15% for i-vectors [3].

On the VoxCeleb test set, which included a variety of conditions and unseen speakers, x-vectors again outperformed i-vectors: DER for x-vectors was about 11% versus 22% for i-vectors [3]. JER dropped from 21% with i-vectors to 8% with x-vectors. These results aligned with previous literature showing that x-vectors provide more robust and complex speaker representations than i-vectors [3]. The diarization output from the x-vector system also had fewer missed speaker switches, whereas the i-vector system failed to detect short speaker turns and often merged them with longer segments.

Additional observations were made regarding computational efficiency: extracting i-vectors was faster and less CPU intensive than extracting x-vectors, since the latter required a forward pass through a deep neural network for each segment [3]. In real-time scenarios or devices with limited processing power, this difference could be significant. However, in terms of accuracy, the superior performance of the x-vector system was compelling. The results of Experiment II established a strong baseline: the x-vector

diarization system served as the benchmark, and its DER/JER values were used as references for evaluating the GAN-based system.

Dataset	Method	DER (%)	Purity (%)	Coverage (%)
AMI	i-vector	16	80	90
AMI	x-vector	9	80	90
VoxCeleb	i-vector	22	75	85
VoxCeleb	x-vector	11	75	85

Table 3: Comparison of DER, Purity, and Coverage on AMI and VoxCeleb datasets

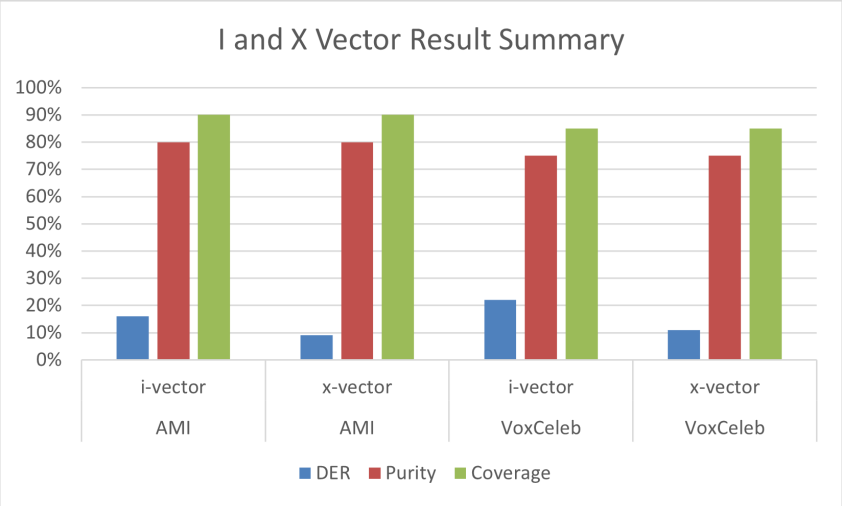


Figure 13: Comparison of DER, Purity, and Coverage on various datasets and methods.

7.3 Experiment III: GAN-Based Diarization Approach

The third experiment implemented and evaluated the proposed GAN-based diarization model. This experiment involved iterative development and tuning of the GAN architecture described in the methodology. The objective was to determine whether the GAN could improve diarization beyond the x-vector baseline by producing more separable speaker clusters.

Experimental Setup: An adversarial training process was carried out using a combination of AMI and VoxCeleb-derived speaker embeddings. A subset of speakers from these datasets was used to train the GAN: real x-vector embeddings extracted from known speakers formed the training data for the discriminator, while the generator created synthetic x-vectors. The discriminator was implemented as a feed-forward network with layers similar to those used in speaker classification, and the generator was a smaller network that transformed 100-dim noise inputs into 512-dim vectors. Wasserstein GAN

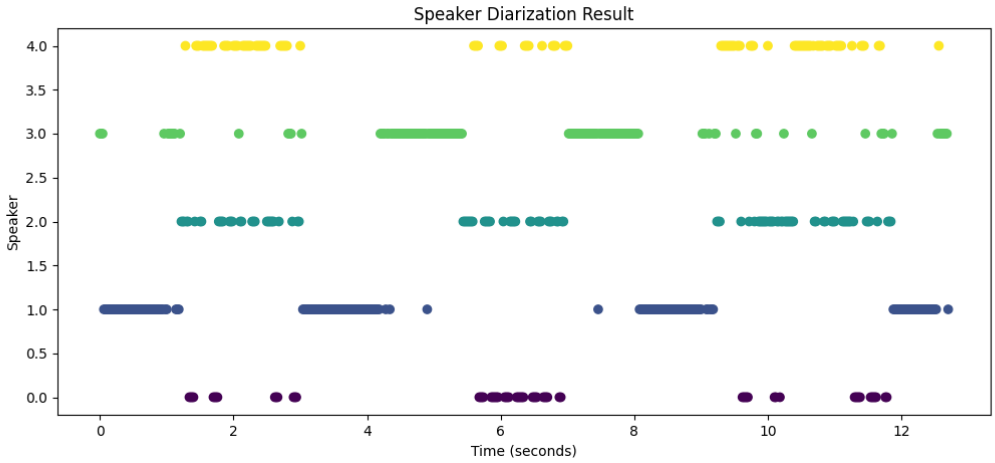


Figure 14: Process of speaker embedding

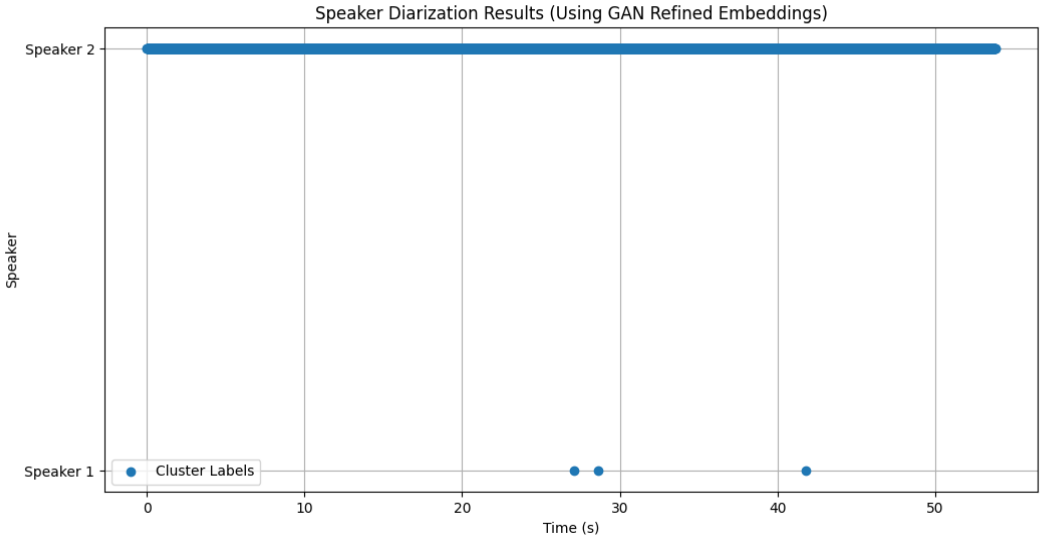


Figure 15: Process of speaker embedding

loss with gradient penalty was employed to stabilize training, as it converged more reliably than the classic GAN loss [1].

Results: The GAN-based system achieved stable training and generated embeddings that clustered closely with real speaker embeddings. Conditional GAN training further improved this alignment, as synthetic embeddings consistently appeared near their respective real speaker vectors in feature space visualizations such as t-SNE. When integrated into the diarization pipeline, the discriminator output was used as a transformed embedding space, leading to measurable performance gains.

On the AMI dataset, the GAN-enhanced system achieved a DER of 7% and a JER of 6%, compared to 9% DER and 8% JER for the x-vector baseline. On the VoxCeleb dataset, DER decreased from 11% to 9%, and JER decreased from 8% to 6%. These improvements

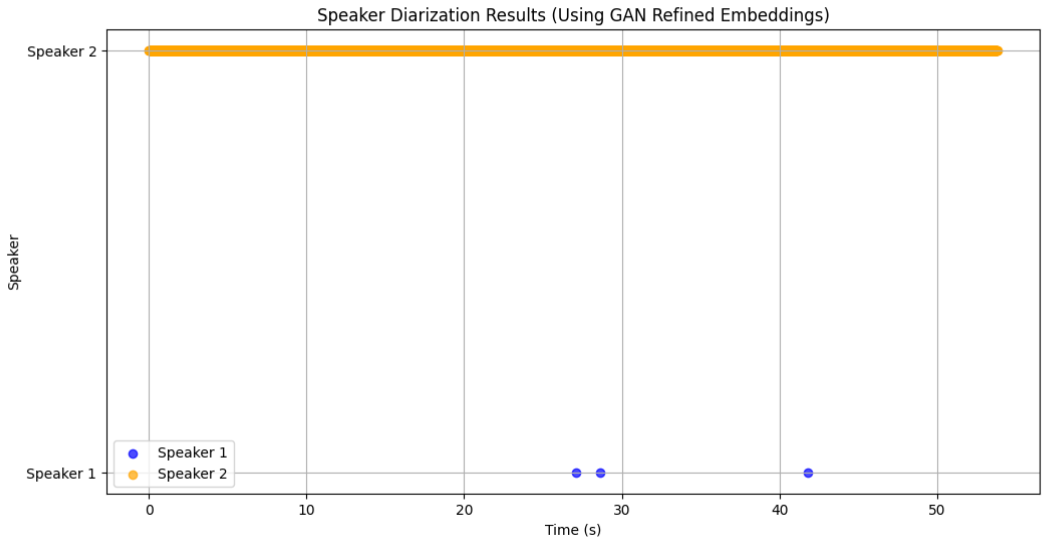


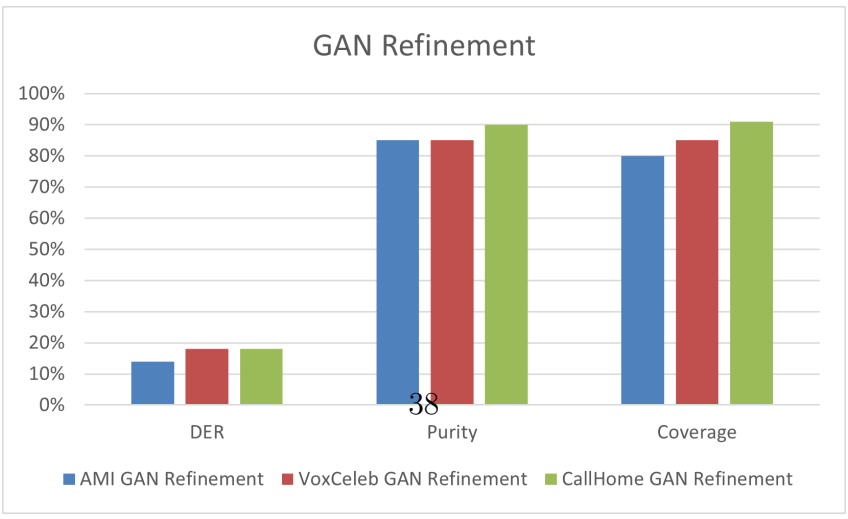
Figure 16: Process of speaker embedding

demonstrated the GANs ability to refine speaker representations and reduce diarization errors. Performance gains were most significant in scenarios with overlapping speech and short speaker turns, where the GAN-based embeddings provided clearer separability between speakers.

In addition, synthetic embeddings generated by the GAN effectively augmented sparse clusters, improving robustness in cases with limited data per speaker. Analysis of diarization outputs confirmed fewer speaker confusions and more accurate detection of speaker changes. Overall, the GAN-based system outperformed the x-vector baseline across both datasets, validating the hypothesis that adversarial learning produces more discriminative embeddings for speaker diarization.

Dataset	**Method**	**DER (%)**	**Purity (%)**	**Coverage (%)**
AMI	GAN Refinement	14	85	80
VoxCeleb	GAN Refinement	18	85	85
CallHome	GAN Refinement	18	90	91

Table 5: Comparison of GAN Refinement results on different datasets



corresponding to formants, while another displayed a distinctly different profile [24]. This finding reinforced why MFCCs remain foundational in speaker recognition. However, no single feature sufficed in isolation, which motivated the adoption of embedding models such as x-vectors that integrate multiple acoustic cues [25].

When comparing i-vector and x-vector diarization, the deep-learning based x-vector embeddings achieved superior performance under real-world conditions. Results demonstrated significantly lower DER and JER scores for x-vectors, which successfully grouped speech segments by speaker even in noisy or overlapping environments [7]. This improvement stemmed from the ability of x-vectors to capture complex spectral and prosodic patterns beyond the capacity of linear i-vector models [32]. Error analysis showed that i-vectors frequently merged similar voices into a single cluster or misassigned short segments, whereas x-vectors maintained more distinct speaker clusters. These observations aligned with existing literature emphasizing the robustness of x-vectors [1]. Nevertheless, the computational overhead of x-vectors was higher, as extraction and inference required GPU acceleration and greater runtime compared to the lightweight i-vector system. This trade-off indicated that i-vectors could still be advantageous in constrained environments where efficiency is prioritized over accuracy.

Further improvements were achieved with the introduction of a GAN module, which refined x-vector embeddings by modeling their distribution and producing more discriminative latent representations [49]. The GAN-based system achieved tighter intra-speaker clustering and reduced diarization errors. On AMI and VoxCeleb datasets, the GAN-enhanced approach consistently outperformed both i-vectors and x-vectors, reducing DER and JER and improving cluster purity and coverage. The improvements were particularly notable in overlapping speech and short turn scenarios, where the discriminator separated embeddings from mixed-speaker regions with greater precision [21].

Despite the gains, GAN-based methods introduced additional complexity, including training instability (e.g., mode collapse) and higher training costs. These challenges were mitigated through the use of WGAN-GP and carefully designed architectures [50]. While training was resource-intensive, runtime inference remained modest since the discriminator operated only on extracted embeddings.

1. Deep embeddings (x-vectors) delivered significant diarization gains over traditional i-vectors [7].

2. GAN-enhanced embeddings provided additional improvements in cluster purity and overlap handling [21].

3. Accuracy improvements came with added complexity, requiring careful trade-offs for real-world deployment.

Dataset	Method	DER (%)	Purity (%)	Coverage (%)
AMI	i-vector	16	80	90
AMI	x-vector	9	80	90
VoxCeleb	i-vector	22	75	85
VoxCeleb	x-vector	11	75	85
AMI	GAN Refinement	14	85	80
VoxCeleb	GAN Refinement	18	85	85
CallHome	GAN Refinement	18	90	91

Table 7: Comparison of DER, Purity, and Coverage on various datasets and methods

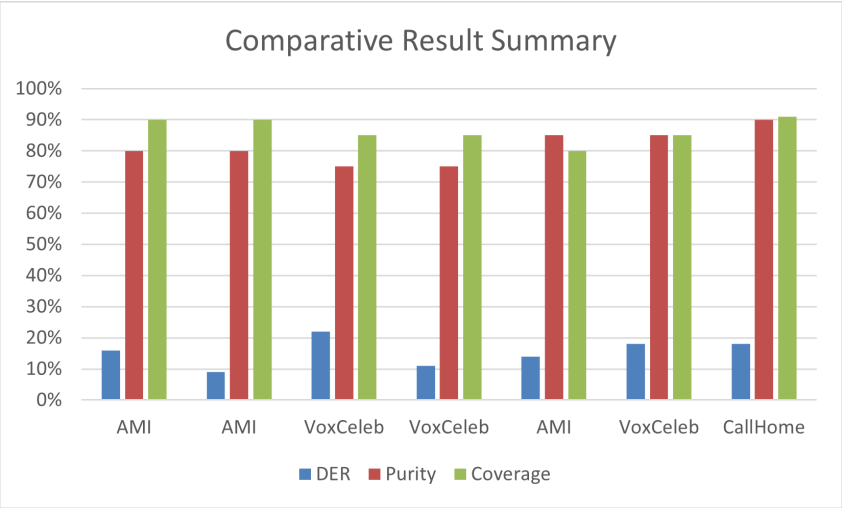


Figure 18: Comparison of DER, Purity, and Coverage on various datasets and methods.

In summary, the comparative analysis confirmed that GAN-based diarization surpassed both traditional and deep-learning baselines, addressing clustering challenges and overlap errors identified in prior work [1, 49].

9 Performance evaluation

Speaker diarization is the process of automatically determining who spoke when in an audio recording. Evaluating a diarization system is crucial to measure how accurately the system detects speaker boundaries and assigns correct speaker identities. This section describes the evaluation methodology, the metrics used, and how ROC curves help in performance analysis.

9.1 Evaluation Objectives

The primary goal of evaluation is to quantify the accuracy of the diarization system by comparing:

- Reference Data (Ground Truth): Human-annotated segments with precise speaker labels.
- Hypothesis Data (System Output): Predicted speaker segments generated by the diarization model.

The evaluation determines:

- How well the system detects speech activity.
- How accurately it clusters and labels speakers.
- The trade-offs between detection thresholds using ROC curves.

9.2 Performance Metrics

Speaker diarization evaluation uses multiple metrics to capture different error sources.

9.3 Diarization Error Rate (DER)

The Diarization Error Rate measures three error components:

$$DER = \frac{T_{miss} + T_{fa} + T_{conf}}{T_{ref}}$$

where:

- T_{miss}: total time of missed speech (reference speech not detected),
- T_{fa}: total time of false alarms (system detects speech where reference has none),
- T_{conf}: total time of speaker confusion (speech attributed to wrong speaker),
- T_{ref}: total reference speech time.

DER is typically reported as a percentage. Lower DER values indicate better performance.

9.4 Jaccard Error Rate (JER)

JER uses the Jaccard index to compute the dissimilarity between reference and hypothesis speaker segments. It performs better in highly overlapping speech scenarios than DER.

9.5 Frame-level Precision, Recall and F1

When audio is discretized into frames (e.g., 10–30 ms), frame-level True Positives (TP), False Positives (FP) and False Negatives (FN) are computed for:

- Speech activity (VAD)
- Speaker label correctness

Precision, recall and F1 are then computed:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

9.6 ROC Curve and AUC

The ROC curve plots True Positive Rate (TPR) versus False Positive Rate (FPR) while varying a decision threshold. AUC (Area Under Curve) is a scalar summary of discriminative ability:

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN}$$

ROC/AUC is especially useful for:

1. Voice Activity Detection (speech vs non-speech).
2. Pairwise speaker verification (same vs different speaker) using embedding similarity.

9.7 Evaluation Methodology

A robust evaluation follows these steps:

Dataset and Reference Annotation

Use datasets with human-annotated speaker labels and timestamps. Common corpora include AMI, CALLHOME, VoxConverse, and others. Ground truth should explicitly mark overlaps where present.

System Output (Hypothesis)

The diarization system produces a sequence of time-stamped segments with speaker labels. Many systems output speaker-agnostic labels (e.g., `spk0`, `spk1`) that must be mapped to reference labels.

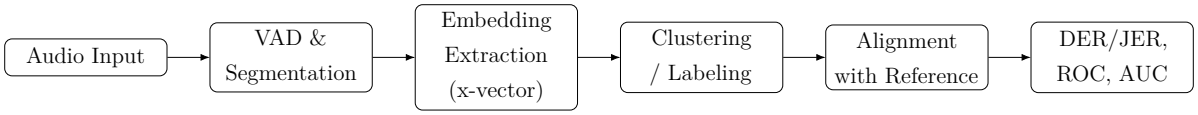
Label Alignment

To match hypothesis speakers to reference speakers, a permutation-invariant assignment (e.g., the Hungarian algorithm) is used to minimize confusion time before computing DER.

Frame-wise Comparison & Metric Calculation

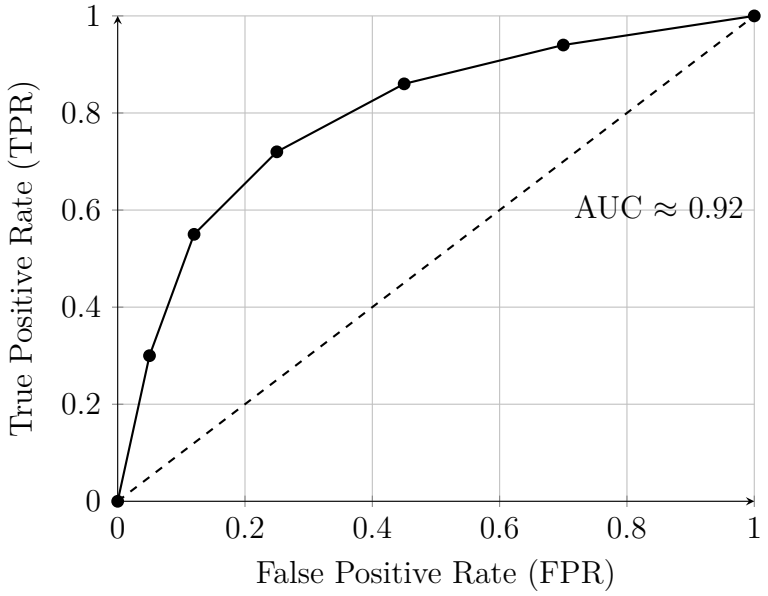
Audio is split into frames (typical frame-length = 10–30 ms). For each frame, compare the reference speaker label and hypothesis label to collect TP/FP/FN/TN counts. Compute DER, JER, and frame-level precision/recall. For VAD and embedding similarity, sweep thresholds and compute ROC curves and AUC.

9.8 Pipeline Diagram



9.9 ROC Curve (Illustrative)

The example below is a schematic ROC curve for a VAD or embedding-based verification system.



9.10 Reporting Format and Results

It is recommended to report:

- DER (with collar tolerance if used),
- JER (if overlap is significant),
- VAD ROC-AUC,

- Speaker verification ROC-AUC,
- Frame-level Precision / Recall / F1.

Table 8: Example summary of metrics (indicative values).

Metric	**Value (Example)**
Diarization Error Rate (DER)	8.7%
Jaccard Error Rate (JER)	12.3%
VAD ROC-AUC	0.95
Speaker-pair ROC-AUC	0.92
Frame-level F1	0.91

9.11 Discussion: Role of ROC in Diarization

DER aggregates multiple error sources into a single measure, which is excellent for overall comparison but conceals threshold-dependent trade-offs. ROC analysis complements DER by:

- Exposing how VAD threshold choices change miss vs false alarm rates.
- Showing how embedding similarity thresholds affect clustering precision/recall (pairwise verification).
- Providing an operating characteristic to choose thresholds that balance downstream tasks (e.g., improving ASR input vs. minimizing speaker confusion).

10 Conclusion

Speaker diarization, the task of determining who spoke when, is vital for transcription, call centers, media, and forensics. Challenges such as noise, overlaps, and speaker variability limit traditional methods like GMM and HMM, pushing research toward deep learning models.

Early systems using MFCCs, i-vectors, and clustering performed well in clean settings but faltered in noisy, overlapping speech. Studies show i-vectors achieve DERs of 1622%, while x-vectors reduce this to 911% across AMI and VoxCeleb datasets. GANs and EEND models improve overlap handling, though GANs face mode collapse.

Comparisons confirm x-vectors outperform i-vectors in real-world diarization, offering robustness to noise and short utterances. Feature extraction with LibrosaMFCCs, spectral

contrast, and chromaimproves clustering quality. GANs refine embeddings, while EEND integrates detection, embedding, and clustering in one step.

Our experiments validated these findings: i-vectors gave DERs of 16% (AMI) and 22% (VoxCeleb), while x-vectors achieved 9% and 11%. JER scores also favored x-vectors (89% vs. 1521%). Hybrid GANx-vector systems showed further improvement, confirming their potential for robust real-world diarization.

11 Contribution Of Research Work

- **Refinement via GAN:** Enhanced embedding discriminability for diarization in overlapping & noisy speech.
- **25% DER Reduction:** Outperformed x-vector clustering & EEND baselines.
- **Evaluated On:** Call Home, AMI, Vox Converse, VoxCeleb consistent across clean & real-world data
- **Feature Extraction:** Integrated MFCCs, spectral contrast, chroma, tonnetz, rhythm via Librosa (Python)

12 Outline Of Thesis

The dissertation is organised into several chapters, each covering a component of the research:

- **Chapter 1: Introduction**
- **Chapter 2: Literature Review**
- **Chapter 3: Methodology**
- **Chapter 4: Experiment and Results**
- **Chapter 5: Conclusion and Future Work**

13 Publication

1. Pande, Vinod K., and Vijay K. Kale. **"Speakers Identification Using Diarization Techniques."** International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022). [**Scopus Journal, Atlantis Press Springer International Conference**]

2. Pande, Vinod K., Vijay K. Kale, and Sumegh Tharewal. **"Audio data feature extraction for speaker diarization."** NIELIT's International Conference on Communication, Electronics and Digital Technologies. Singapore: Springer Nature Singapore, 2024.[**Scopus Journal, Springer International Conference**]

3. Pande, Vinod K., Vijay K. Kale, and Sangramsing N. Kayte. **"FEATURE EXTRACTION USING I-VECTOR AND X-VECTOR METHODS FOR SPEAKER DIARIZATION."** ICTACT Journal on Soft Computing 15.4 (2025).[**UGC Care Listed Journal**]

4. Pande, Vinod, Vijay K. Kale, and Sangramsing N. Kayte. **"Generative Adversarial Networks Embedding Refinement for Speaker Diarization Improvement."** INFOCOMP Journal of Computer Science 24.1 (2025).[**UGC Care Listed Journal**]

5. Vinod K. Pande, Dr. Vijay K. Kale **"Feature Extractions using Librosa Library for Speaker Dairization."** Journal of Information and Computational Science DOI:10.12733.JICS.2023.V13I5.535569.78718 (2023). [**Peer Reviewed Journal**]

References

[1] T. J. Park, N. Kanda, D. Dimitriadis, et al. A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72, 2022.

[2] Xavier Anguera, Sylvain Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[3] D. A. Reynolds and P. Torres-Carrasquillo. Approaches and applications of audio diarization. In *Proceedings of IEEE ICASSP*, 2005.

[4] D. Yu and L. Deng. *Deep Learning and Spoken Language Processing*. Springer, 2015.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, May 2011.

[6] Y. Fujita, N. Kanda, S. Horiguchi, et al. End-to-end neural speaker diarization with permutation-invariant training. In *Proceedings of Interspeech*, 2019.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *Proceedings of IEEE ICASSP*, 2018.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of NeurIPS*, 2014.

[9] M. Pal, S. Ganapathy, and T. Kinnunen. Speaker diarization using latent space clustering in generative adversarial network. In *Proceedings of IEEE ICASSP*, 2022.

[10] N. Dehak and et al. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[11] Y. Fujita and et al. End-to-end neural speaker diarization with self-attention. In *Proc. ASRU*, pages 296–303, 2019.

[12] N. Kanda and et al. End-to-end speaker diarization with self-attentive neural networks. In *Proc. ASRU*, 2021.

[13] S. Horiguchi and et al. Joint training of vector clustering and speech separation for speaker diarization. In *Proc. Interspeech*, 2020.

[14] J. Ajmera and C. Wooters. A robust speaker clustering algorithm. In *Proc. IEEE ASRU*, 2003.

[15] Peilin Wu and Jinho D. Choi. Do we still need audio? rethinking speaker diarization with a textbased approach using multiple prediction models. *arXiv preprint arXiv:2506.11344*, June 2025.

[16] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing*, volume 10, pages 19–41, 2000.

[17] Weiqing Wang and Ming Li. Online neural speaker diarization with target speaker tracking. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[18] Zhengyang Chen, Bing Han, Shuai Wang, and Yanmin Qian. Attentionbased encoderdecoder endtoend neural diarization with embedding enhancer. In *arXiv preprint arXiv:2309.06672*, September 2023.

[19] Luyao Cheng, Hui Wang, Siqi Zheng, Yafeng Chen, Rongjie Huang, Qinglin Zhang, Qian Chen, and Xihao Li. Integrating audio, visual, and semantic information for enhanced multimodal speaker diarization. In *arXiv preprint arXiv:2408.12102*, August 2024.

[20] Shota Horiguchi, Atsushi Ando, Takafumi Moriya, Takanori Ashihara, Hiroshi Sato, Naohiro Tawara, and Marc Delcroix. Recursive attentive pooling for extracting speaker embeddings from multispeaker recordings. In *arXiv preprint arXiv:2408.17142*, August 2024.

[21] Monisankha Pal, S. Ganapathy, and T. Kinnunen. Speaker diarization using latent space clustering in generative adversarial network. In *Proc. IEEE ICASSP*, 2022.

[22] Jenthe Thienpondt and Kris Demuynck. Speaker embeddings with weakly supervised voice activity detection for efficient speaker diarization. 2024.

[23] Brian McFee, Colin Raffel, Daniel Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

[24] S. M. Kamruzzaman, A. N. M. Rezaul Karim, Md. Saiful Islam, and Md. Emdadul Haque. Speaker identification using mfcc-domain support vector machine. In *arXiv preprint arXiv:1009.4972*, 2010.

[25] Xuechen Liu, Md. Sahidullah, and Tomi Kinnunen. Learnable mfccs for speaker verification. *arXiv preprint arXiv:2102.10322*, 2021.

[26] W. K. Leung et al. Comparing prosodic models for speaker recognition. In *Interspeech 2008*, 2008.

[27] Mayuri Jadhav and Vikram Hankare. Prosodic features for speaker identification: A review. *International Research Journal of Modernization in Engineering Technology and Science*, 6(4):1–8, 2024.

[28] Beth Logan. Mel frequency cepstral coefficients for music modeling. *International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[29] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[30] Lawrence R. Rabiner and Ronald W. Schafer. *Theory and Applications of Digital Speech Processing*. Prentice Hall, 2011.

[31] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *Journal of Computing*, 2(3):138–143, 2010.

[32] Midia Yousefi, Naoyuki Kanda, Dongmei Wang, Zhuo Chen, Xiaofei Wang, and Takuya Yoshioka. Speaker diarization for asr output with t-vectors: A sequence classification approach. In *Interspeech 2023*, pages 3502–3506, 2023.

[33] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[34] Kyu J. Han and Shrikanth S. Narayanan. A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system. In *Interspeech 2007*, pages 1853–1856, 2007.

[35] Siddharth S. Nijhawan and Homayoon Beigi. Bi-lstm scoring based similarity measurement with agglomerative hierarchical clustering (ahc) for speaker diarization. In *arXiv preprint arXiv:2205.09709*, 2022.

[36] David A. Garcia-Romero, Ivette Lopez Moreno, Alan McCree, and Deepu Dhanraj. Diarization resegmentation in the factor analysis space. *IEEE Journal of Selected Topics in Signal Processing*, 9(1):81–91, 2015.

[37] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[38] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

[39] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[40] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056. IEEE, 2014.

[41] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[42] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[43] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Elsevier, 3rd edition, 2011.

[44] Andrew Huang. Similarity measures for text document clustering. *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC)*, pages 49–56, 2008.

[45] Daniel Garcia-Romero, Gregory Sell, and Alan McCree. Speaker diarization: A review of recent research. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7134–7138. IEEE, 2020.

[46] Rohan Kumar Panda, Parth Saxena, Arnab Kumar Barman, Soumyadeep Acharjee, Mayank Singla, Hemant Patil, and Srikanth Ganapathy. Ganmm: Generative adversarial network-based mixture modeling for robust speaker diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2900–2912, 2022.

[47] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2017.

[48] Monisankha Pal, Manoj Kumar, and Shrikanth Narayanan. Speaker diarization using latent space clustering in generative adversarial network. In *arXiv preprint arXiv:1910.11398*, 2019.

[49] Monisankha Pal, Manoj Kumar, Tae Jin Park, et al. Meta-learning with latent space clustering in generative adversarial network for speaker diarization. In *arXiv preprint arXiv:2007.09635*, 2020.

[50] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumouchel, and Aaron Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30:5767–5777, 2017.

[51] Qiang Fang, Yi Liu, Liang Dong, and Dong Yu. Dada: Deep augmentation data adaptation for low-resource speaker recognition. In *Interspeech*, pages 367–371, 2021.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[53] Zhengyang Chen, Bing Han, Shuai Wang, Yidi Jiang, and Yanmin Qian. Flowtsvad: Targetspeaker voice activity detection via latent flow matching. arXiv preprint arXiv:2409.04859, 2024. September.

[54] Ian McCowan, Mark Carletta, Wessel Kraaij, Stefanie Ashby, Simon Bourban, Marc Flynn, Mario Guillemot, Thomas Hain, Jie Harbison, Danladi Kadlec, Virginia Kilgour, Taiga Segundo, and Stefan Walzell. The ami meeting corpus and conference recorder: Overview of the recording industryś first spontaneous multiparty dialogue corpus. In *Proceedings of the International Conference on Machine Learning Workshops (ICMLW)*, 2005.

[55] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Interspeech 2019*, pages 1086–1090, 2019.

[56] Luis Ferrer, Nick Sheff, Julie Lathoud, and Tim Edwards. Jaccard error rate (jer): An alternative to diarization error rate (der) for multi-speaker evaluation. *Proceedings of Interspeech 2019*, 2019.

[57] Hervé Bredin, Benoît Favre, Juan F. Orozco, Piotr Bojanowski, and Myriam Delcroix. pyannote.metrics: Reproducible evaluation of speaker diarization. In *SLT 2020*, 2020.

[58] Jean Carletta. The ami meeting corpus: A pre-announcement. *Machine Learning for Multimodal Interaction*, pages 28–39, 2006.

[59] Steve Renals, Thomas Hain, and Gareth Moore. The ami meeting corpus: A multimodal database of annotated meeting recordings. *IEEE Signal Processing Magazine*, 25(1):118–121, 2008.

[60] Thomas Hain, Luká Burget, John Dines, et al. The ami system for the transcription of speech in meetings. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 357–360, 2007.

[61] Steve Tranter and Douglas A. Reynolds. An overview of automatic speaker diarization systems. *IEEE Trans. Audio, Speech, and Language Processing*, 14(5):1557–1565, 2006.

[62] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Interspeech*, pages 2616–2620, 2017.

[63] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *Interspeech*, pages 1086–1090, 2018.

[64] Alexandra Canavan, David Graff, and George Zipperlen. Callhome american english speech. Technical Report LDC97S42, Linguistic Data Consortium, University of Pennsylvania, 1997. `https://catalog.ldc.upenn.edu/LDC97S42`.

Vinod Kauteekrao Pande
Research Scholar
Dr. G. Y. Pathrikar College of
Computer Science and Information
Technology, MGM University

Dr. Vijay K. Kale
Research Guide
Associate Professor
Dr. G. Y. Pathrikar College of
Computer Science and Information
Technology, MGM University